Energy Aware Performance Evaluation of WSNs

Krishna Chaitanya Doddapaneni

School of Science and Technology

Middlesex University

A thesis submitted to Middlesex University in fulfilment of the requirements for the degree of

Doctor of Philosophy

February 24, 2015

To Dhana Lakshmi - Narayana Rao and Karthik

everything I write has a precedent in truth

Abstract

Distributed sensor networks have been discussed for more than 30 years, but the vision of Wireless Sensor Networks (WSNs) has been brought into reality only by the rapid advancements in the areas of sensor design, information technologies, and wireless networks that have paved the way for the proliferation of WSNs. The unique characteristics of sensor networks introduce new challenges, amongst which prolonging the sensor lifetime is the most important. Energy-efficient solutions are required for each aspect of WSN design to deliver the potential advantages of the WSN phenomenon, hence in both existing and future solutions for WSNs, energy efficiency is a grand challenge. The main contribution of this thesis is to present an approach considering the collaborative nature of WSNs and its correlation characteristics, providing a tool which considers issues from physical to application layer together as entities to enable the framework which facilitates the performance evaluation of WSNs. The simulation approach considered provides a clear separation of concerns amongst software architecture of the applications, the hardware configuration and the WSN deployment unlike the existing tools for evaluation. The reuse of models across projects and organizations is also promoted while realistic WSN lifetime estimations and performance evaluations are possible in attempts of improving performance and maximizing the lifetime of the network. In this study, simulations are carried out with careful assumptions for various layers taking into account the real time characteristics of WSN.

The sensitivity of WSN systems are mainly due to their fragile nature when energy consumption is considered. The case studies presented demonstrate the importance of various parameters considered in this study. Simulation-based studies are presented, taking into account the realistic settings from each layer of the protocol stack. Physical environment is considered as well. The performance of the layered protocol stack in realistic settings reveals several important interactions between different layers. These interactions are especially important for the design of WSNs in terms of maximizing the lifetime of the network.

Acknowledgements

I am grateful to everyone who has been there to support my journey towards the finished thesis and contributed in the work it stands for. I hereby want to thank all of them, and apologize that I will not be able to name every one of them.

A prominent place and heartfelt thanks are owed to my director of studies Prof. Orhan Gemikonakli, my supervisors Dr. Enver Ever and Dr. Leonardo Mostarda for their outstanding supervision in every stage of my Ph.D. I could not have asked for better role models, each inspirational, supportive, patient, immensely knowledgeable and helpful. I have enjoyed the opportunity to watch and learn from their experience and leadership qualities. Their frequent insights and patience with me are always appreciated.

I would have never made it to Middlesex for my PhD in the first place if it hadn't been for my Masters thesis supervisor, Dr. Enver Ever, to whom I will forever be thankful. His humility, friendliness made it easier to bond with him. I greatly value the close personal rapport that Dr Ever and I have forged over the years. His broad horizon of knowledge and infectious enthusiasm towards work benefited me a lot through this important period of my life. He is a person of a great character, and has been a selfless mentor.

I am thankful to Dr. Leonardo Mostarda for inspiring and encouraging me to visit and work with scholars from various universities on several different occasions, a truly exciting opportunity. Your expertise and guidance has been of unlimited value to me.

Dr Purav Shah has been extremely supportive and influenced my research immensely. For the countless hours spent discussing fruitful ideas over cups of coffee, for the pleasure and the privilege to discuss WSN research issues, for the cricket we played, I would like to express my profound gratitude.

I am deeply honoured and feel very fortunate to have worked with Professor Ali Tasiran, without his support, collaboration and guidance, my doctoral research would not have been the same.

I owe a great debt of gratitude to my teachers at BVBPS, Hyderabad. I could not be prouder of my academic roots and hope that I can in turn pass on the research values and the dreams that they have given to me.

I am indebted to all my family in London, who helped me in numerous ways during various stage's of my Ph.D.

I must thank my colleagues and friends at Middlesex, L'Aquila and Camerino for their generous support. They have been highly influential on my life.

I am lucky to have had a wonderful group of close friends who are most encouraging and have helped me through the last few years, in particular Yoney, Priti, Amit, Niharika, Yonal, Pragya, Unai, Nallini, Pawel, Ryan, Ammar, Fred, Emmanual, Sindhu,....

Above and beyond all, my heartfelt gratitude to my parents and brother for their much needed support, patience, understanding, and encouragement in every possible way. You made me into who I am.

For every action towards graduation there is an equal and opposite distraction. The path to becoming a Doctor is littered with distractions. I'd like to thank those distractions for making me the person I am.

I am sure I have missed many others, although not intentionally; I thank all of you.

List of Publications

The work presented in this thesis has given rise to the following publications.

- Krishna Doddapaneni, Ali Tasiran, Fredrick A. Omondi, Enver Ever, Purav Shah, Leonardo Mostarda, and Orhan Gemikonakli: Does the Assumption of Exponential Arrival Distributions in Wireless Sensor Networks Hold?, International Journal of Distributed Sensor Networks, 2014 (under review).

- Ivano Malavolta, Leonardo Mostarda, Henry Muccini, Krishna Doddapaneni, Enver Ever, Orhan Gemikonakli: A4WSN: An Architecture-driven Modelling Platform for Analysing and Developing WSNs. (Journal of Systems and Software 2014: Under Review)

- Krishna Doddapaneni, Enver Ever, Orhan Gemikonakli, Ivano Malavolta, Leonardo Mostarda, Henry Muccini: A model-driven engineering framework for architecting and analysing Wireless Sensor Networks. SESENA 2012: 1-7

- Krishna Doddapaneni, Enver Ever, Orhan Gemikonakli, Ivano Malavolta, Leonardo Mostarda, Henry Muccini: Path Loss Effect on Energy Consumption in a WSN. UKSim 2012: 569-574

- Krishna Doddapaneni, Enver Ever, Orhan Gemikonakli, Leonardo Mostarda, Alfredo Navarra: Effects of IDSs on the WSNs Lifetime: Evidence of the Need of New Approaches. TrustCom 2012: 907-912

- Krishna Doddapaneni, Omondi Fredrick, Enver Ever, Purav Shah, Orhan Gemikonakli, and Gagliardi Ricardo: Deployment Challenges and Developments in Wireless Sensor Networks Clustering, in Proceedings of The 28th IEEE international Conference on Advanced Information Networking and Applications (AINA 2014)

- Sule Clifford, Purav Shah, Krishna Doddapaneni, Orhan Gemikonakli,and Enver Ever, On demand Multicast Routing in Wireless Sensor Networks, in Proceedings of, The 28th IEEE international Conference on Advanced Information Networking and Applications (AINA 2014)
- Krishna Doddapaneni, Purav Shah, Enver Ever, Ali Tasiran, Fredrick A. Omondi, Leonardo Mostarda, and Orhan Gemikonakli: Packet Arrival Analysis in Wireless Sensor Networks, The 29th IEEE international Conference on Advanced Information Networking and Applications (AINA 2015)

Contents

List of Figures

List of Tables

Chapter 1

Introduction

"The most profound technologies are those that disappear. They weave themselves into the fabric of everyday life until they are indistinguishable from it".

- The late Mark Weiser, *Father of Ubiquitous Computing*

Wireless Sensor Networks (WSNs) have attracted a wide range of disciplines where close interaction with the physical world are essential. The past several years have seen an interesting increase in the development of WSNs. With recent advancements in wireless communications as well as Micro Electro Mechanical Systems (MEMS) technology, the implementation of low cost, low power multifunctional sensor nodes, smaller in size and communicate untethered in short distances have become feasible. A sensor node is a small digital device consisting of sensor(s), a microprocessor, a transceiver and a power source, with very limited sensing, processing and communication capabilities. A sensor network is composed generally of a large number of sensor nodes, densely deployed either inside or close to the physical phenomenon. The tiny sensor nodes can act as both data generators and network relays. These unique characteristics and intrinsic properties of individual sensor nodes and WSNs separate them from other communication networks, and also present unique challenges for the development of communication protocols in terms of energy consumption as the stringent energy reserves of the sensor nodes make the energy consumption of primary importance.

WSNs, with a wide range of applications are rapidly becoming an integral part of our lives. Over the last decade, WSNs have appeared as one of the most prominent enabling technologies of MEMS, which combines automated sensing, embedded computing, and wireless capabilities into tiny devices, bringing promises of understanding and instrumenting nature at scales that were unimaginable before [Akyildiz et al., 2002a; Emary and Ramakrishnan, 2013]. Recently, considerable amount of research efforts have enabled the actual implementation and deployment of sensor networks tailored to the unique requirements of certain sensing and monitoring applications. The application of sensor networks are diverse, ranging from habitat monitoring to surveillance and physical intrusion detection and can be categorised into environment, health, military, home, disaster relief, space exploration and other commercial areas. The flexibility, fault tolerance, low cost, rapid deployment characteristics and high sensing fidelity of sensor networks create many new and exciting applications in the field of remote sensing. WSN applications and communication protocols are tailored mainly to provide higher energy efficiency, as sensor nodes carry limited power sources. Energy efficiency is crucial because of the scale and application environments in which sensors are deployed [Akyildiz and Vuran, 2010; Langendoen et al., 2014].

Performance modeling and evaluation should consider metrics for WSNs, such as system lifetime and energy efficiency, and the introduction of new traffic attributes, which enables to evaluate WSNs in a better way.

1.1 Performance Evaluation Techniques

A lot of research in computer science and electrical engineering is being carried out in the field of distributed systems and computer networks. Topics being addressed include the development of new and improved protocols, applications, architectures, security and quality-of-service techniques, just to name a few. A crucial step during the design and engineering of communication systems, respective protocols, algorithms and architectures, is the estimation of their performance, and the understanding and visualization of the micro and macro behavior of the systems and their components. Typically, this can be (more or less) realized by applying three different methodologies: (1) experiments with real systems and prototypes, (2)

mathematical analysis, and (3) simulation [Banks, 2010; Jain, 1991]. Performance and availability analysis of the systems becomes very essential for the success or failure of many projects. The existing studies consider benchmarking in the form of test beds and measurements for real deployment. The energy constrains of WSNs, limits their processing capabilities and communication. Therefore, using one of these performance evaluation methods, and analysis of deployment and management of such complex systems is a challenging task [Akyildiz et al., 2002c].

The process of performance analysis by the means of actual measurements is benchmarking and the measurements and input workloads used are the benchmarks. Benchmarking is possible only when something similar to the proposed system already exists. Hence, benchmarking gives very accurate results. However, these results in most of the cases are of limited use since extrapolation of these results to suit the changes in the system or environment is usually not possible. Also, the benchmarking studies are usually expensive in terms of the cost of equipment as well as the time required to set-up and test various configurations, especially for large scale systems such as WSNs with large numbers of nodes [Jain, 1991].

Analytical modelling techniques abstract the features of a parallel system as a set of parameters or parametrized functions, in order to make the modelling task tractable. Due to inherent complexity and diverse nature of WSNs (dynamic topology, wireless channel characteristics, mobility, density of the nodes etc.), analytical modelling sometimes requires a degree of assumptions to simplify the systems considered [Banks, 2010; Jain, 1991; Trivedi, 2002]. These simplifications may lead to inaccurate results in case of unrealistic assumptions [Chen et al.; Krop et al., 2007]. Analytical models generally provide the best information for the effects of various parameters and their interactions [Jain, 1991]. Often, analytical methods show the borderline behaviour of system characteristics or offer upper and lower bounds for specific research questions. However, more fine grained analysis often leads to an unacceptable complexity of the analytical models.

Simulation studies mimic the operation of a real world process, over time [Banks, 2010]. In contrast, simulations offer scientists and researchers a controlled environment in which a system can be investigated in more detail. Different parameter sets and scenarios can be analysed with comparably little effort. Thus, they are considered as a powerful and versatile methodology to analyse and visualize the

behaviour and performance of communication systems and networks, hence, are currently the most widely adopted method for analysing WSNs. They also provide quicker evaluation, optimisation and modifications of the proposed algorithms and protocols at design, development and implementation stages. Simulation models can be validated against the existing systems and then can be altered to reflect the proposed modifications. A number of simulation tools are available with different features, models, architectures and characteristics for performance evaluation in WSNs. Simulation studies are very flexible providing fairly accurate and acceptable results, however, for sufficient accuracy, simulations require relatively high computational times [Law and Kelton, 2000]. The goal of performance studies are to compare various alternatives and/or to find the best architecture in terms of performance and cost effectiveness and to find the optimal parameter values [Ever, 2007]. Simulation studies are widely used in computer science for performance, availability and reliability evaluation and analysis of complex computer and communication systems.

1.2 Scope of Investigation

The unique characteristics of WSNs introduce new challenges, amongst which prolonging the sensor lifetime is the most important, hence making energy consumption a major design parameter. Sensor networks are constrained in terms of memory and processing power, making protocol stacks with large memory footprints undesirable. Similarly, interfaces that require high processing between layers cannot be supported. In addition, the broadcast and non-deterministic nature of the wireless channel creates interdependencies between each layer, especially with the low-power communication techniques. Recent empirical studies motivate that the properties of low-power radio transceivers and the wireless channel conditions should be considered in protocol design. Finally, the event-centric approach of WSNs requires application aware communication protocols. The main drawback of the protocols developed for WSNs is that they follow the traditional layered protocol architecture. While they may achieve very high performance in terms of the metrics related to each of the individual layers, they are not jointly optimized to maximize the overall network performance while minimizing energy expenditure.

These unique characteristics of WSNs have motivated cross-layer protocols that include the functionalities of two or more layers in a single coherent framework. Recent studies on WSNs reveal that cross-layer integration and design techniques result in significant improvement in terms of energy conservation. The existing layered communication protocols improve the energy efficiency to a certain extent by exploiting the collaborative nature of WSNs and its correlation characteristics. Considering the scarce energy and processing resources of WSNs, joint optimization and design of networking layers, i.e., cross-layer design, stands as the most promising alternative to inefficient traditional layered protocol architectures.

Realization of sensor networks needs to satisfy various constraints introduced by factors such as topology change, wireless channel characteristics, environment, power consumption, cost, hardware etc. as these constraints are highly stringent and specific for sensor networks, separating them from other communication networks. Radio interference, whether intentional or otherwise, represents a serious threat to assuring the availability of sensor network services. Moreover, recent empirical studies necessitate that the properties of low power radio transceivers and the wireless channel conditions be considered in protocol design [Ganesan et al., 2008; Güngör and Hancke, 2013; Zuniga and Krishnamachari, 2009]. Energy-efficient solutions are required for each aspect of WSN design to deliver the potential advantages of the WSN phenomenon. Therefore, in both existing and future solutions for WSNs, energy efficiency is the grand challenge. **This research project presents an approach considering the collaborative nature of WSNs and its correlation characteristics, providing a tool which considers issues from physical to application layer together as entities to enable the A4WSN framework. The simulation approach considered provides a clear separation of concerns amongst software architecture of the applications, the hardware configuration and the WSN deployment unlike the existing tools for evaluation. The reuse of models across projects and organizations is also promoted while realistic WSN lifetime estimations and performance evaluations are possible in attempts of improving performance and maximizing the lifetime of the network.** In this study, simulations are carried out with careful assumptions for various layers taking into account the real time characteristics of WSN.

A4WSN was developed in collaboration. I worked on simulation engine side of the tool. Although the A4WSN tool is built to be independent, the modules that should be included requires extensive research and understanding of WSN systems. Furthermore unlike the existing simulation approaches, since path loss effects are incorporated. Although a variety of outdoor path loss models exist, we were in need for a path loss calculation approach for indoor environment to be applicable for WSNs. I was also responsible from deriving and applying appropriate path loss models for WSNs. The path loss details are in turn translated to provide data that can be fed to the A4WSN. The design and optimization of a wireless sensor network draws on knowledge and understanding of many different areas such as properties of the radio front end (which determine what type of MAC protocols can be used), the type of application (which limits the options for routing protocols), sleeping patterns of a node, security mechanisms employed, etc. all influence the performance of WSN. Wireless channel is a dominant factor in the performance of communication protocols in WSNs. The low-power communication capabilities of sensors and the rather limited capabilities of low-cost transceivers result for a significant impact on higher layers. Hence, the effects of the wireless channel cannot be confined only to the physical layer. Most of the existing approaches overlook the effects of path loss resulting in overestimation of resources. The tool presented in this study provides a more realistic approach for evaluation of WSNs particularly in presence of path-loss.

The networks topology can be defined as the location of the nodes that are available for communication, according to the graph theory. The topology of the network should be able to guarantee certain requirements such as connectivity, coverage, or lifetime [Abbasi and Younis, 2007; Jardosh and Ranjan; Khan et al., 2013; Kumarawadu et al., 2008; Li, 2006; Santi, 2005]. The resulting topology directly affects not only the performance of each individual protocol, but also the overall performance of the network. Clustering techniques for WSNs have been extensively studied and they have proven to improve the network lifetime, a primary metric, used for performance evaluation of sensor networks. Although introduction of clustering techniques has the potential to reduce energy consumption and extend the lifetime of the network by decreasing the contention through either power control or node scheduling, scalability remains an issue. In this study, the

clustering protocols are considered in presence of a new tool with separation of concerns, various environmental details etc.

The MAC protocols ensure communication in the wireless medium such that, the communication links are established between the nodes and connectivity is provided throughout the network. Moreover, the access to the channel should be coordinated in a way that, collisions which constitute a major source of energy consumption in WSNs, are either minimized or eliminated [Akyildiz and Vuran, 2010]. When considering energy issues in WSNs, various issues present in data link layer must also be taken into account, due to the energy critical nature of the network.

With the ever growing capabilities of WSNs, the need for security becomes apparent. Due to their unattended nature and limited resources, they pose new security challenges. Although research efforts have been made in areas such as cryptography, key management, secure routing, secure data aggregation, and Intrusion detection, these services also add more computation, communication and storage overhead in WSN, and hence consume more energy. Especially in case of WSNs, the matter becomes very delicate as the energy consumption can have severe effects on the networks lifetime while the network is being protected. Accordingly, it is necessary to assess the performance trade-off's of the cryptographic algorithms to provide a better understanding of the security cost. If wireless sensor networks support a variety of security functions, more powerful devices and more energy are required. This is mainly because, providing more advanced security features would significantly affect the energy consumption of the overall network. Incorporating network layer details along with physical layer (combining path loss information) with the MAC layer design improves performance in WSNs.

The sensitivity of WSN systems are mainly due to their fragile nature when energy consumption is considered. The reliable operation of a wireless sensor network is closely tied to the ability of the sensor radios to successfully communicate with each other. The tool presented, considers various issues from physical to application layer such as path loss, various clustering techniques, size of the cluster, security related mechanisms into account for performance evaluation in terms of maximizing the life time of the network. The tool presented also provides a more realistic approach for evaluation of WSNs particularly in presence

of path-loss. The case studies presented demonstrate the importance of various parameters considered in this study. Simulation-based studies are presented for all the issues considered. The performance of the layered protocol stack in realistic settings reveals several important interactions between different layers. These interactions are especially important for the design of WSNs in terms of maximizing the lifetime of the network.

1.3 Aims and Objectives

The main focus of the thesis is to evaluate WSNs in a more realistic way. The objectives include:

- The main contribution of the thesis is to present an approach considering the collaborative nature and correlation characteristics of WSNs.
- The tool presented considers issues from physical to application layer together as entities to enable the presented framework.
- The simulations are carried out with careful assumptions for various layers taking into account the real time characteristics of WSNs.
- The framework presented provides a clear separation of concerns amongst software architecture of the application, the hardware components and the WSN deployment considered.
- These interactions are especially important for the design of WSNs in terms of realistic life time estimations and performance evaluations and for maximising the lifetime of the network.

1.4 Outline of the Thesis

Chapter 2 introduces the domain of the research by providing a critical review of the related literature. Existing work on WSNs architecture and a sensor node is presented in detail. The protocol stack used by the sensor nodes is also critically analysed. An overview of the ever increasing applications of WSNs are also

presented. They include existing commercial applications developed for WSNs. Various constraints introduced by factors such as topology change, wireless channel characteristics, environment, etc. have deep impact on the performance of the WSNs as these constraints are highly stringent and specific for sensor networks, separating them from other communication networks. All the factors are critically analysed in this chapter. The need for a new approach in-order to overcome challenges such as abstraction, separation of concerns and reuse is also presented along with the existing work in the literature.

One of the common configurations to prolong the lifetime and deal with the path loss phenomena is having a multi-hop set-up with clusters and cluster heads to relay the information. Although researchers continue to address these challenges, the type of data arrival distributions at the cluster head and intermediary routing nodes is still an interesting area of investigation. The general practice in published works is to compare an empirical exponential arrival distribution of wireless sensor networks with a theoretical exponential distribution in a Q-Q plot diagram. In chapter 3, we show that such comparisons based on simple eye checks are not sufficient since, in many cases, incorrect conclusions may be drawn from such plots. After estimating the Maximum Likelihood parameters of empirical distributions, we generate theoretical distributions based on the estimated parameters. To the best of our knowledge, this is the first work that provides statistical proof for finding theoretical distributions of arrivals at the CH and relay nodes in WSNs. By conducting Kolmogorov-Smirnov Test Statistics for each generated inter-arrival time distributions, we find out, if it is possible to represent the traffic into the cluster head by using theoretical distribution. Empirical exponential arrival distribution assumption of wireless sensor networks holds only for a few cases. There are both theoretically known such as Gamma, Log-normal and Mixed Log-Normal of arrival distributions and theoretically unknown such as non-Exponential and Mixed cases of arrival distributions in wireless sensor networks. The work is further extended to understand the effect of delay on inter-arrival time distributions based on the type of medium access control used in wireless sensor networks.

Despite the ever increasing usage of WSNs in modern applications, their development is still plagued by many issues. These issues are to be addressed, hence ameliorating the implementation and enabling different analysis in terms of various

factors, such as energy efficiency and performance evaluation to be performed at the early stages of WSN design and development. In Chapter 4, a rich multi-view modelling environment has been proposed, supported by a powerful programming framework, for the model-driven engineering of wireless sensor networks. The modelling viewpoints and conceptual elements have been carefully designed, including the domains such as software engineering, wireless sensor networks, simulations and telecommunications. The programming frameworks functioning has been tested by realizing a plug-in devoted to energy and performance related simulations of WSNs.

In order to validate the expressivity of the A4WSN modelling languages and to exercise the provided extension points, an analysis plug-in called PlaceLife has been developed, and is presented in Chapter 5. An overview on the existing simulators is also presented. This work shows that when path loss is introduced, increasing the transmission power is needed to reduce the amount of lost packets. This presents a trade-off between the residual energy and the successful transmission rate when more realistic settings are employed for simulation. In order to show the usefulness and effectiveness of the approach presented, a case study based on home automation system is also presented. Numerical results along with the analysis of various factors affecting the performance in terms of energy consumption of WSNs are given.

Since it is likely that the data acquired from one sensor node is highly correlated with the data gathered from its neighbours, the redundant information can be reduced with the help of data aggregation. All of the sensor nodes collaborate together to form a communication network for providing reliable networking service. Thus, node clustering, which aggregates nodes into groups (clusters), is critical to facilitate practical deployment and operation of WSNs. In Chapter 6, the major issues and challenges in node clustering for WSNs are discussed and a variety of state-of-the-art clustering techniques are introduced and discussed in detail. The affect of path loss on well known clustering protocols are also presented, encouraging realistic WSN lifetime estimations and performance evaluations. Also, the bottlenecks in the network, in terms of cluster size scalability, especially while addressing variety of high packet sending rate and real-time applications, such as wearable heart rate and physical activity monitors and holster monitors is pre-

sented.

Though sensor networks share some commonalities with typical computer networks, the unique requirements of its own make them a special type of network. Wireless sensor networks pose new security challenges because of their unattended nature and limited resources. Intrusion detection in WSN is a particularly challenging task because of the limited resources of the nodes. WSNs can operate in two different modes called as continuous periodic sensing and transmission or event-triggered sensing. The decision on which mode of operation to use is highly dependant on the application. In Chapter 7, the affects of intrusion detection solutions on the lifetime of the WSNs is studied. More specifically, comparisons between approaches that continuously monitor the network and those that use some kind of agreement in order to discover the attackers and isolate them is presented.

Chapter 8 summarises the main contributions of the thesis and outlines some possible avenues for future studies.

Chapter 2

Literature Survey

2.1 Introduction

The design of WSNs requires ample knowledge of a wide variety of research fields including wireless communication, networking, embedded systems, digital signal processing, and software engineering. In this chapter, an overview of sensor nodes and basic system architecture including the network protocol stack are provided along with an overview of representative sensor network applications. The design choices at various layers significantly impact the operation and resource efficiency of sensor nodes and networks. Chapter 2 begins this discussion with an introduction to WSN architecture and concepts. Since the wireless medium is shared between many sensor nodes, MAC-layer protocols are required to arbitrate access to the wireless channels. It also surveys the need for security, multi-hop communications in WSNs and the associated challenges. Moreover, the integration of the solutions for these factors is still a major challenge because of the interdisciplinary nature of this research area. The need of using software engineering approaches in order to support the design, analysis, simulation and implementations of WSNs, which considers the collaborative nature of WSNs along with its correlation characteristics is also presented.

2.2 WSNs Architecture

The wireless sensor nodes are the central element in a WSN. It is through a node that sensing, processing, and communication take place. It stores and executes the communication protocols and the data-processing algorithms. The quality, size, and frequency of the sensed data that can be extracted from the network are influenced by the physical resources available to the node. Therefore, the design and implementation of a sensor node along with network architectures and protocols are important aspects in the design of wireless sensor networks [Akyildiz et al., 2002c]. Hence, it is quite important to understand the architecture of WSNs while considering them for energy aware performance evaluation, as they are quite different from the traditional OSI model stack. A typical sensor network architecture consists of a sensor field (physical environment), where the sensor nodes are usually scattered as shown in Figure 2.1. The sink/gateway communicates with the end user via wireless network. The low-cost sensor nodes are capable of being both data originators and data routers. Data is routed back to the end user either in single or multi-hop fashion, through the sink. The protocol stack used by the sensor nodes is given in 2.2. The protocol stack consists of the *application layer*, *transport layer*, *network layer*, *data link layer*, *physical layer* along with *power management plane*, *mobility management plane*, and *task management plane*. This protocol stack combines the power and routing awareness, while integrating the data and networking protocol, power efficiently communicating through the wireless medium. In this section, we first introduce various layers of a typical sensor network and then describe the architecture of a sensor node.

Physical Layer

It is a commonly acknowledged fact that the properties of the transmission channel and the physical-layer shape the significant parts of the protocol stack for WSNs. The physical layer deals with the modulation and demodulation of the data, along with the responsibility of frequency selection, signal detection, and data encryption. Hence, it is a challenging task to find simple, low cost modulation schemes and architectures of the transceiver. Wireless communication provide

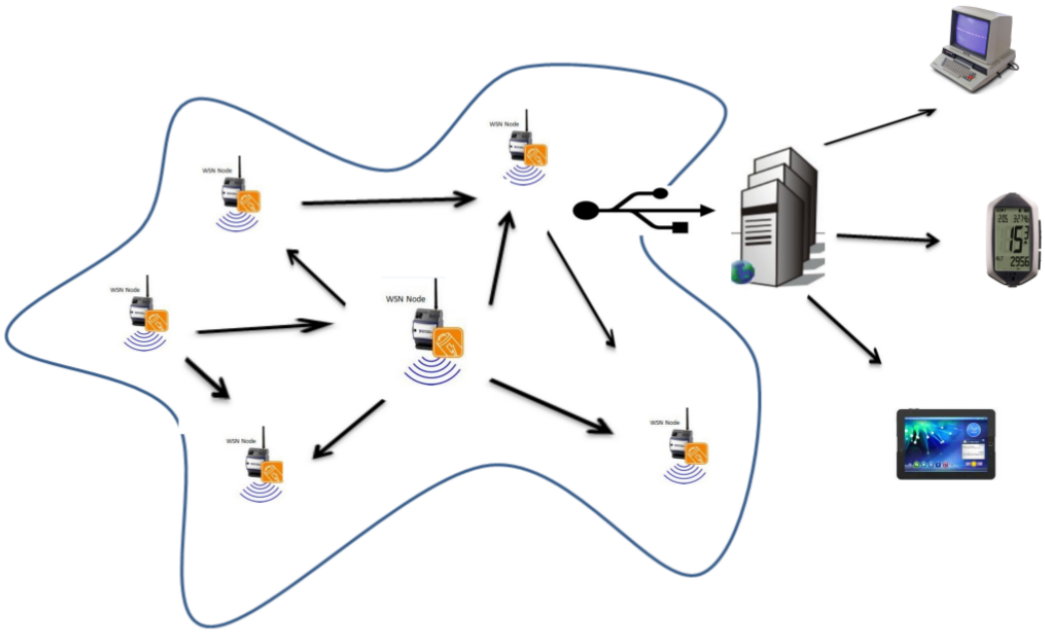

Figure 2.1: Typical sensor networks architecture

most of the unique advantages like ease of deployment, broadcast communication etc., along with several challenges such as limited communication range, interference and frequent errors [Akyildiz and Vuran, 2010; Karl and Willig, 2007]. Most of the existing studies do not consider scattering, reflection, refraction, shadowing, decay, multi-path effects etc. together with energy efficiency.

Data Link Layer

The data link layer is responsible for multiplexing of the data streams, medium access, error control and data frame detection, and ensuring reliable point-to-point connections in a communication network. The low cost requirements and distributed nature of the sensor nodes restrict the energy consumption of the data link layer [Akyildiz et al., 2002a]. Therefore, for the design of MAC layer, energy efficiency is of primary importance as MAC constitutes the core of the communication coordination. The MAC dictates the states of the radio. Power consumption of the sensor is based on the time the radio is on (either listening, transmitting, or receiving) i.e. the state of the radio and the amount of time it stays in each

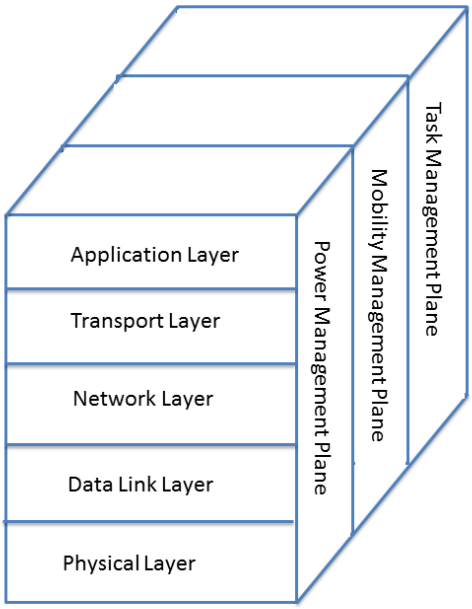

Figure 2.2: Generic protocol stack for WSN

of these states. The MAC protocol should ensure that sensor nodes provide energy efficient operations in WSNs, as energy wastage in sensor networks occurs due to the fact that there are collisions, wasted awake times when sensor nodes are listening to the channel but not receiving anything, unnecessary overhearing and also due to the fact that wireless medium suffers from multipath and fading that causes a serious threat to reliable data transmission over the communication link, leading to retransmissions. The MAC protocols ensure communication in the wireless medium such that the communication links are established between the nodes and connectivity is provided throughout the network [Akyildiz and Vuran, 2010].

Network Layer

Due to the inherent characteristic, routing in WSNs is a very challenging task. Many challenging factors influence the design of routing protocols. The routing challenges and design issues that affect the routing process in WSNs include

connectivity, coverage, data aggregation, Quality of service (QoS), deployment of the nodes, energy consumption without loosing accuracy, fault tolerance, scalability, mobility, transmission media, etc [Al-karaki and Kamal, 2004]. Based on the network structure, routing protocols for WSNs can be categorised into *flat-based* routing, where all the nodes are typically assigned equal functionality or roles, *hierarchical-based* routing, where nodes play different roles in the network and *location-based* routing, where the sensor nodes positions are exploited to route the data in the network. These protocols can be furthermore classified, depending on protocol operation into *multipath-based*, *query-based*, *negotiation-based*, *QoS-based* or *coherent-based* routing techniques.

Transport Layer

End-to-end reliability and congestion control are the two main functionalities of the transport layer protocols. The success and efficiency of WSNs directly depend on the reliable communication between the sensor nodes and the sink [Akyildiz and Vuran, 2010]. Due to the influence of the hardware constraints such as limited power and memory, the development of transport layer protocols is a challenging task, resulting in inability to store large amounts of data and acknowledgements which can be quite costly sensor networks. Transport layer protocols should be able to mitigate congestion, that may occur due to high traffic. Limited resources and high energy costs also prevent end-to-end reliability mechanisms being employed in WSNs. Due to the processing, storage and energy limitations in sensor nodes, transport layer protocols should aim to exploit the collaborative capabilities of the resource constrained sensor nodes and shift the intelligence from the sensor nodes to the sink [Karl and Willig, 2007].

Application Layer

High sensing fidelity, flexibility, low cost, fault tolerance and rapid deployment characteristics of WSNs create many new and exciting applications, making sensor networks an integral part of our lives. The application layer provides the user with necessary interfaces to interact with the physical environment through the WSN. The application layer includes several network management functionalities along

with the main application [Akyildiz and Vuran, 2010]. The unique characteristics of WNSs have motivated cross-layer protocols, with the functionalities of two or more layers in a single coherent framework, for achieving significant improvement in terms of energy conservation [Fang and McDonald; Van Hoesel et al., 2004].

Protocol stack

The protocol stack used in sensor nodes contains physical, data link layer, network, transport and application layers, and can be summarised as follows [Wang et al., 2006]:

- Physical layer: deals with frequency selection, carrier frequency generation, modulation, signal deflection, and data encryption.
- Data link layer: deals with medium access, multiplexing of data streams, error control, ensuring reliability in data transfer.
- Network layer: deals with assignment of addresses and forwarding packets.
- Transport layer: responsible for specifying how the reliable transporting the packets.
- Application layer: deals with data requests and interactions with the end user.

2.2.1 Architecture of a Sensor Device

A sensor device primarily has a sensing unit that performs the actual sensing tasks (e.g. detecting changes in light, temperature etc.), a processing unit which is responsible for performing computations on the sensed data in conjunction with a storage unit as well for handling communication with other sensor nodes, and a power management unit and responsible for reducing the power consumption in the sensor node. Figure 2.3 shows a simple schematic of the internal architecture of a sensor node [K.Pahlavan and P.Krishnamurthy, 2009]. The processing unit contains the microprocessors which processes information received from the

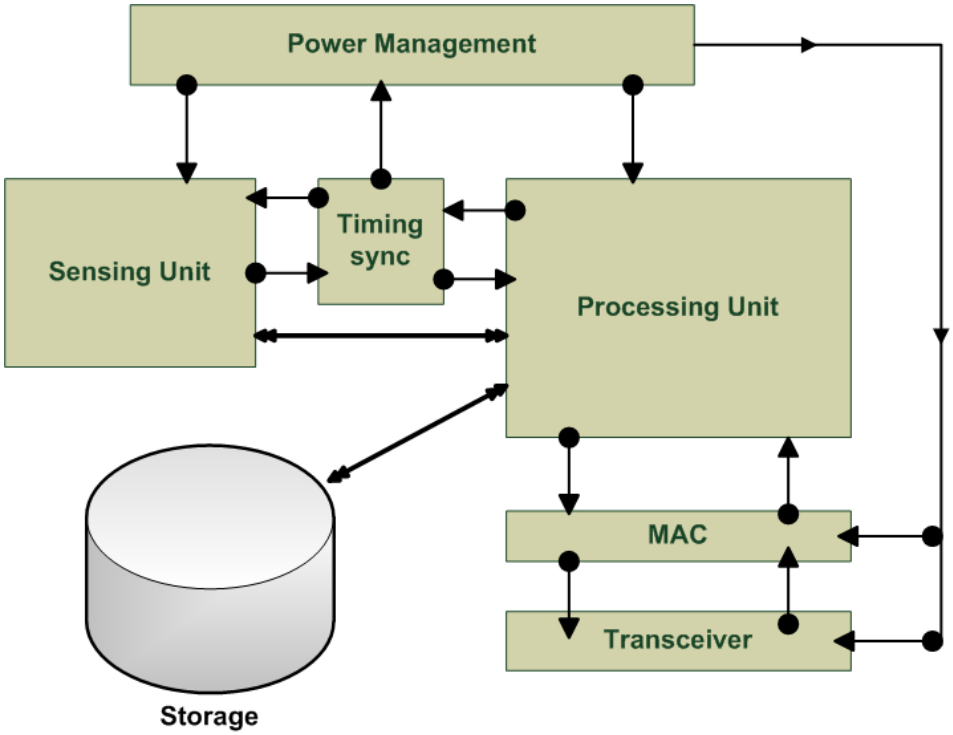

Figure 2.3: Sensor node architecture

sensing unit. The processing unit performs partial processing of data due to energy constraints, while the sink node performs the extensive processing of data. The communication unit consists of a transceiver, customised to operate in a resource constrained environment. The modular design approach provides flexible and versatile platform to address the needs of a wide variety of applications.

2.3 Basic Models

The applications and corresponding traffic characteristics in WSNs are very different from those of the traditional networks. For example, the widely used applications for Internet include e-mail, Web-based services, the file transfer protocol, and peer-to-peer services, WSNs have totally different ones. As a result, traffic and data delivery models are also different. The data delivery model has a significant impact on the performance of energy-efficient routing algorithms for WSNs, so approaches which are very advantages for proactive networks do not

work well with reactive ones, and vice versa. Currently, four traffic models are used in WSNs: event based delivery, continuous delivery, query-based delivery, and hybrid delivery. Traffic model greatly influences protocol design and affects performance, especially in energy consumable areas. Optimum outcome from the developing protocol cannot be achieved if appropriate data delivery model is not chosen [Tilak et al., 2002]. The four models and the related performance aspects are discussed below.

- The data collected by the sensors need to reported regularly, perhaps continuously, or periodically. For example, on Great Duck Island, a WSN is used to observe the breeding behaviour of a small bird. Time based medium access control protocols can be used to achieve significant energy savings in case of continuous data delivery models.

- Most event-driven applications in WSNs are interactive, delay intolerant (real-time), mission critical and non-end-to-end applications. In this case, sensor nodes monitor the occurrence of events passively and continuously. The query-driven data delivery model is very similar to the event-driven model, except that the data is pulled by the sink where as in event-driven models, the data is pushed to the sink. CSMA medium access arbitration is a good fit for event-based data delivery models since the data is generated sporadically.

- Similar to event-driven applications, most query-driven applications in WSNs are also interactive, mission critical, query-specific, delay tolerant and non-end-to-end applications. Queries can be sent on demand in order to save energy. Sometimes, the sink may be interested in a specific piece of information that has already been collected by the sensor nodes. The sensor only reports the observed data in response to an explicit request from the user. Query-driven systems store gathered information locally and communicate it on request. This type of sensor network can be useful in logistics or home applications, but is not very common in applications of environmental Monitoring.

- Some networks apply a hybrid model using a combination of continuous, event-driven and query-driven data delivery, as the types of sensors and the data they sense may be very diverse. Examples include, data to be reported continuously by some nodes, and the sink may need to query information from other sensor nodes.

2.4 Applications of WSNs

The emergence of WSN paradigm has triggered extensive research, with emphasis on potential applications that can be realized using WSNs. Most of the time, the behaviour of the sensor network highly depends on the applications, within a specific environment. Sensor networks offer a powerful combination of distributed sensing, computing and communications. Though they pose formidable challenges due to their peculiarities, they lend themselves to various countless applications, virtually in all fields of science and technology and hence making their way to the forefront of the scientific community [Arampatzis et al., 2005; Puccinelli and Haenggi, 2005].

Although sensor network research is initially driven by military applications such as battlefield surveillance and tracking, high-end applications such as radiation and nuclear threat detection, sensor networks now are widely deployed in diverse applications including home automation, environmental monitoring, microsurgery, robotics, support for logistics, agriculture etc. [Arampatzis et al., 2005; Puccinelli and Haenggi, 2005]. The ever increasing applications of WSNs can be categorised as follows:

Wireless Sensor Network Applications				
Military	Environmental	Health	Home	Industrial
Smart Dust	Great Duck Island	Artificial retina	Water monitoring	Preventive monitoring
Sniper detection	CORIE	Patient monitoring	Home automation	Structural monitoring
Surveillance	ZebraNet	Emergency response	Fire alarm system	VigilNet
VigilNet	Volcano Monitoring	Code Blue		
	Flood Detection			

Military Applications

The rapid deployment, self organization and fault tolerant characteristics of sensor networks make them a promising technique for military systems, and an integral part of military command, control, communications, computing, intelligence, surveillance, reconnaissance and targeting (C4ISRT) systems [Akyildiz et al., 2002a]. Military applications of sensor networks include battlefield surveillance, where critical terrains, approach routes etc. can be rapidly covered with sensor networks and can be closely monitored. With the evolving new plans of operation, new sensor networks can be deployed for surveillance purposes such as:

- Monitoring friendly forces, equipment and ammunition: sensor nodes models and architectures have been researched and deployed successfully for monitoring friendly forces, equipment and ammunitions as it is essential for friendly forces to prevent their base, equipment and ammunition from being attacked [Lee et al., 2009].
- Targeting: For intelligent ammunitions, sensor networks can be incorporated into guidance systems, and other military applications also include battle damage assessment [Lee et al., 2009].
- Nuclear, biological and chemical attack detection and reconnaissance [Akyildiz et al., 2002a].

Healthcare

The medical applications of WSNs aim to improve the existing health care and monitoring facilities. Recent technological advancements in implanted biomedical devices and smart integrated sensors have opened up new prospects for a variety of healthcare systems and biomedical applications [Alwan et al., 2006; Lorincz et al., 2004; Shnayder et al., 2005]. WSN implementations on pervasive computing based health care systems avoid various limitations and drawbacks associated with the wired sensors, providing a better-quality of care, quicker diagnosis, more intense collection of information (can be employed for statistical analysis) and at

the same time keeps the cost and resource utilization to minimal. Monitoring facilities introduced by using WSNs are particularly useful for early detection and diagnosis of emergency conditions, as well as keeping track of the diseases for patients. WSN based health care systems are also useful for providing a variety of health related services for people with various degrees of cognitive and physical disabilities [Alemdar and Ersoy, 2010]. In [Alwan et al., 2006], the authors present a pilot study conducted in collaboration with the volunteers of America National Services. A number of In-home Monitoring Systems (IMS) were deployed in an assisted living setting. Similarly “CodeBlue", introduced in [Malan et al., 2004], is a wireless sensor infrastructure for seamless transfer of data among caregivers, and efficient allocation of hospital resources. Some of the health care applications include emergency response, provision of interfaces for disabled, tracking and monitoring doctors and patients inside a hospital, integrated patient monitoring, diagnostics, tele-monitoring of physiological data and drug administration [Akyildiz and Vuran, 2010; Gao et al.].

Environmental Applications

Environmental applications of WSNs include precision agriculture, environmental observations and forecasting weather phenomena, tracking the movements of birds and small animals, forest fire detection, meteorological or geophysical research, flood detection, pollution studies etc. [Akyildiz and Vuran, 2010; Basha et al., 2008; Cerpa et al., 2001; Juang et al., 2002]. Sensors networks are extremely useful for monitoring physical phenomena over large geographical areas, by deploying large number of sensors. One good example of habitat monitoring is the Great Duck Island project, to study the distribution and abundance of sea birds on Great Duck Island [Mainwaring et al., 2002]. Other habitat monitoring projects include CORIE [201, 2013] and ZebraNet [Juang et al., 2002], and environmental projects including volcano monitoring [Werner-Allen et al., 2006], flood detection [Bonnet et al., 2000].

Home Applications

Home control applications provide control, conservation, convenience and safety to the end user, allowing the user to manage home devices locally and remotely [Sohraby et al., 2007]. The smart sensors and actuators can be buried into appliances such as heating, ventilation and air conditioning (HVAC), vacuum cleaners, microwave ovens, televisions, refrigerators, as well as to water monitoring systems. These smart sensors allows the users to manage home devices locally as well as remotely, enabling the interconnection of various devices at residential places with convenient control of various applications at home [Akyildiz and Vuran, 2010].

Civil and Structural Health Monitoring

Research has been under way in recent years to develop sensor technology that is applicable for buildings, bridges, tunnels and other structures. Structural health monitoring refer to periodic or continuous monitoring of the health of these large structures, and detecting the changes in structures that effect their performance [Kim, 2005; Xu et al., 2004]. Potential damage can be localized and the extent of damage can be estimated in almost real time. The main physical quantities to measure are internal material stress, temperature, moisture, displacements and tilting at a fine-grained level. Some potential application also include environmental monitoring in coal mines [Li and Liu, 2007]

2.5 WSN Topologies

The network topology of WSNs affects the network connectivity and organization and various performance metrics such as communication, networks scalability, reliability, data latency, energy efficiency and network life time, etc. Therefore, current research depicts customized domain-specific WSN topologies for efficient utilization of their constrained resources. Topology structure plays the an important role in designing and constructing WSNs.

Star, tree, mesh and clustered hierarchical architectures have emerged as popular choice topologies in WSNs [Shrestha and Xing, 2007]. A star network is a communication topology where a single base station can send and/or receive a

message to a number of remote nodes. The remote nodes cannot send messages to each other; they can only send or receive a message from a single base station. The advantage of star network for wireless sensor networks is the ability to keep the remote nodes energy consumption to a minimum due to its simplicity. It also allows for low latency communications between the remote node and the base station. The disadvantage of a star network is that the base station must be within radio transmission range of all the individual nodes and is not as robust as other networks due to its dependency on a single node to manage the network

Mesh networks are multi-hop local area networks in which the node connectivity is arbitrary. Each sensor node not only sends and receives data, but also acts as router to relay messages for its neighbours in the network, hence facilitating multiple communication paths from the sensor nodes to the base station. A mesh network allows for any node in the network to transmit to any other node in the network within its radio transmission range. Redundancy and scalability are the advantages of Mesh networks. If an individual node fails, a remote node still can communicate to any other node in its range, and makes the communication possible. Also, the range of the network is not necessarily limited by the range of a single node; it can be extended by adding more nodes to the system. The disadvantage of a Mesh network is the power consumption for the nodes. Due to multi-hop communication, the battery of the nodes close to the base station drains quickly, often limiting the life of the network. Also, as the number of communication hops increases, the time to deliver the message also increases, especially if low power operation of the nodes is a requirement.

In order to support data aggregation, to minimise the total number of messages exchanged between nodes and hence to save energy through efficient network organization, nodes can be partitioned into a number of smaller groups called clusters. Each cluster has a coordinator, known as a Cluster Head (CH), and a number of member nodes which communicate only to their CH in order to transmit data. Clustering results in a two-tier hierarchy in which CHs form the higher tier and member nodes form the lower. Clustering offers some advantages such as data aggregation done at the CH level, distribution of load across all nodes since the role of the CH is not permanently fixed to one particular node; hence rotation of CH is present. CH handles two types of traffic: intra-cluster and inter-cluster commu-

nication; the former being communication between member nodes of a cluster and the CH, the latter being the transmission/relay of packets from one CH to another CH, until it reaches the base station. Inter-cluster communication can make use of either single hop or multi-hop forwarding. However, in multi-hop clustering, nodes nearest to the base station tend to deplete their energy the fastest since they are burdened with heavy relay traffic from the rest of the network in addition to their own intra-cluster traffic share. Those nodes closer to the base station tend to die earlier than the rest, and as a result there is reduction in the coverage area of the network and also network partitioning becomes apparent. This is known as hot spot problem [Xuhui et al., 2009; Zhao and Wang, 2010a]. CH selection process can be based on various parameters such as energy resources, distance to the base station, number of nodes etc. Clustering in WSNs faces several serious deployment issues, such as, ensuring connectivity between all the nodes, CH rotation, optimal cluster sizes and hot spot problem [Li et al., 2005a].

Choosing the right topology is very important in order to optimize the performance of WSNs, for specific applications and within a specific environment.

2.6 Security

Security mechanisms in WSNs are essential as they provide data integrity, authentication, confidentiality, user privacy, access control and reliability. However, due to the characteristics that distinguish them from traditional wireless networks such as node deployment, unreliability of sensor networks, severe energy, computation, bandwidth and memory constraints, securing a WSN is a challenging task. Development of fundamental security tools such as broadcast authentication and key management are an important step for protecting sensor networks, as they provide basic building blocks for implementation of various security mechanisms [Liu and Ning, 2007].

- **Broadcast Authentication:** The authenticity of broadcast commands and data is very crucial for sensor network operation, as it is usually desirable for base stations to broadcast commands and data to the sensor nodes, due to the large number of nodes and the broadcast nature of wireless communication,

and forging or modified commands or data might lead to incorrect operations and may fulfil the intended purposes of the network.

- **Key Management:** It is an important fundamental security service, ensuring sensor nodes to communicate securely with each other, with the help of cryptographic techniques. Traditional pairwise key establishment techniques such as public key cryptography and key distribution centre are not always feasible due to the intensive computations involved in signature verification and resource constraints on sensor nodes.

Although prevention measures such as encryption [Jia et al., 2008] and firewalls [Ma et al., 2006] can be used, the attacker can physically access the WSN and tamper with the nodes in order to subvert the correct WSN behaviour. Intrusion Detection Systems (IDSs) can be used to mitigate the problem. They are a second line of defence that analyses the observable behaviour of a system in order to recognise malicious behaviour.

There are two main types of intrusion detection techniques: *misuse* and *anomaly*. Misuse detection systems [T.Eckmann et al., 2002] are explicitly programmed to recognise well-known attacks. These systems recognise intrusions by matching the pattern of observed data with the set of predefined (intrusion) signatures. They can perform focused analysis thus having a low false alarm rate. However, they cannot detect unknown types of attacks. Anomaly detection systems assume that an attack will cause deviation from normal behaviour, thus detection can be done by comparing actual activities with known correct behaviours. Different approaches have been used to model normal behaviour: statistics-based [Javitz and Valdes, 1994], rule-based [Vaccaro and Liepins, 1989] and formal specification [Stillerman et al., 1999]. The advantage of this kind of systems is the ability of detecting unknown attacks. However, it is not easy to define what is a normal behaviour and set up anomaly thresholds in order to have a good detection efficiency and a low positive rate.

Intrusion detection in WSN is a particularly challenging task because of the limited resources of the nodes. WSNs can operate in two different modes called as continuous periodic sensing and transmission or event-triggered sensing. The decision on which mode of operation to use is highly dependant on the application.

For WSNs, while the IDS enhances security, it can shorten the lifetime of the WSN since the IDS may require to run in promiscuous mode [Chen et al., 2007], [Filipovic and Datta, 2004]. More precisely in promiscuous mode, each IDS can continuously eavesdrop the radio in order to check the correct behaviour of all other nodes. This solution not only makes impossible to optimise the duty cycle (nodes can never sleep) but also requires the nodes to be in the same range. However promiscuous mode is not really suitable for applications with event-triggered sensing.

It is quite important not only to consider improving the WSN system in terms of security, but at the same time to consider the over heads that may be caused, and their effects on the WSNs lifetime, while considering the performance evaluation of WSNs.

2.7 Path loss

It is well known that long-distance wireless communication can be expensive, in terms of both energy consumption and implementation complexity. While designing the physical layer for sensor networks, energy minimization assumes significant importance, over and above the decay, scattering, shadowing, reflection, diffraction, multi-path, and fading effects. Because of the behaviour of the Radio Frequency(RF) and potential obstacles within the environment in order to have the optimization studies in a more realistic manner, path loss should also be considered. Calculation of signal coverage is very essential for design and deployment of wireless networks. Signal coverage in wireless networks is influenced by a variety of factors, in which radio frequency and the terrain are the prominent ones. Transmission range is one of the most important parameter for design and operation of WSNs, essentially the distance upto which reliable communication is possible between a transmitter and receiver [K.Pahlavan and P.Krishnamurthy, 2009].

The wireless channel distorts the signals transmitted from a receiver, due to four major phenomena:

- **Attenuation:** The signal strength is attenuated as the signal wave propagates through air. The attenuation is proportional to the distance travelled over the air, resulting in path loss for radio waves in the air. More specif-

ically, there is decrease in signal strength as a function of distance, which also defines the transmission range of a node [Goldsmith, 2005].

- **Reflection and refraction:** A certain fraction of the wave bounces off the surface, when a signal wave is incident at a boundary between two different types of material, known as reflection, and a certain fraction of the wave may also propagate through the boundary, known as refraction. Reflection and refraction are usually observed on the ground or the walls of a building, resulting in fading of the received signal based on constructive or destructive effects of multiple waves, received by the receiver [Goldsmith, 2005].

- **Scattering:** Signal wave scatters, when the signal is incident at a rough surface. This results in multiple copies of the signal being received at the receiver [Goldsmith, 2005].

- **Diffraction:** Signal propagation through sharp edges such as a building or an object. A new wave is generated from the sharp edge,which act as a source, thus effecting the signal strength, as it is distributed to the new waves. In WSNs, due to the short range communications, diffraction is not a major factor in low-power communications. However, factors such as scattering and reflection also affect the wireless channel communication [K.Pahlavan and P.Krishnamurthy, 2009].

Path loss is reduction in signal strength as a function of distance, and is also a consequence of many effects such as free-space loss, refraction, diffraction, reflection, aperture-medium coupling loss, and absorption [Goldsmith, 2005; K.Pahlavan and P.Krishnamurthy, 2009]. Path loss is also affected by other factors such as propagation medium (dry or moist air), the distance between the transmitter and the receiver, and the frequency of the signal [Rappaport, 1996]. Indoor radio propagation is dominated by the same mechanisms as outdoor propagation: reflection, scattering, diffraction, refraction, absorption and depolarization. However, conditions are much more variable. The indoor environment differs widely due to the increased number of obstacles, layout of rooms, presence of multiple walls and floors, windows and open spaces [Goldsmith, 2005]. Altogether these factors have a significant impact on path loss in an indoor environment. Due to the irregularity

in the position of obstacles and layout of the rooms, the channel varies significantly with the environment making the indoor propagation modelling relatively inconsistent and challenging especially for modelling. The propagation and path loss models are usually based on empirical studies on the system considered. Accurate modelling of the actual environment is very complex as the communication systems operate in complex propagation environments. In practice, most of the simulation studies make use of the empirical models that have been developed based on empirical measurements over a given distance in a given operational frequency range and a particular environment [Rappaport, 1996].

Path loss is a non-negligible phenomenon in WSNs. It is an essential reason for asymmetric radio interference as well as asymmetric links in upper layer of the protocol stack. It can directly or indirectly affect many aspects of the performance of upper layers [Zhou et al., 2006]. The effects of path loss are not only confined to the MAC and routing layers, but also influences other protocols such as sensing coverage, localization and topology control protocols.

2.8 MAC Protocols - Collisions

Harnessing the potential benefits of WSNs requires a high-level of coordination, self-organization and management among the sensors to support the underlying application and perform the assigned tasks. Unlike the communication over a guided medium in a wired network, communication in wireless network is achieved in the form of electromagnetic signal transmission through a common transmission medium, which much be shared by all the sensor nodes in a fair manner. Therefore, shared access of the channel requires the establishment of a MAC protocol among the sensor nodes, and hence WSNs performance is highly dependent on the choice of the medium access protocol. The need to preserve energy is the most critical issue in the design of stable and scalable MAC layer protocols for WSNs. Excessive overhead, packet collisions, idle listening and overhearing are the factors that contribute to energy waste. The main objective of most MAC-layer protocols is to reduce the energy waste caused by the above mentioned factors [Lin et al., 2007]. MAC protocols for WSNs can be categorised into two main groups: schedule based and contention based. The main objective of schedule based MAC

protocols is to achieve a high level of energy efficiency in order to prolong the network lifetime, while that of contention-based MAC layer protocols is to minimize, rather than completely avoid, the occurrence of collisions. To reduce energy consumption, these protocols differ in the mechanisms used to reduce the likelihood of a collision while minimizing overhearing and control traffic overhead. One of the key limitations of traditional contention based channel access protocols is that the nodes consume energy needlessly when they are idle as well as when collisions occur. Collisions in WSN are one of the major source of increased latency and packet retransmissions [Rajendran et al., 2003]. Collisions occur when more then one node attempts to transmit simultaneously or when the communicating node assumes that the data are lost due to errors caused by noise on the communication channel, and since corrupted packets must be retransmitted, collisions add an additional burden to the already energy constrained system. The collisions caused on resource constrained WSN leads to extra latency and numerous retransmissions and hence equating to excess energy consumption and network performance degradation (wastage of energy, lower bandwidth utilization and larger data delivery latency) [Lin et al., 2007; Rajendran et al., 2003; Stathopoulos et al., 2004].

Packet success ratio also drops due to frequent collisions and retransmissions, the data glut increases the time delay to reach the sink and the energy consumption in WSN. Data collisions lead to the corruption of data in transmitted packet which has to be discarded and the follow-on retransmissions increase energy consumption as well as the network latency [Hu and Cao, 2010; Lin et al., 2007; Rajendran et al., 2003; Stathopoulos et al., 2004]. The half duplex nature of wireless channel prevents collision detection, therefore, collision avoidance techniques are usually exploited by MAC protocols. In WSNs, the performance optimization studies must consider the effects of collision domains, apart from the factors at different layers such as path loss, the rate of transmission, data aggregation and the right topology. Although, a variety of MAC protocols have been proposed for WSNS, no protocol has been standardised. The primary reason is that sensors networks are application specific, and hence a MAC protocol is usually application dependent. TDMA and CSMA are among the most common underlying MAC protocols that are used for sensor networks. Collision free nature is the major advantage of TDMA, which can significantly improve the energy efficiency of the network even under high traffic.

Due to idle time slots, it has higher delay and low throughput especially under low traffic load. Moreover, TDMA requires strict time synchronization between different sensor nodes, and has limited scalability and adaptability to network changes. In contrast, CSMA are contention based, which results in lower energy efficiency and higher delay under high traffic load, but can reduce delay and has higher throughput under low traffic load. Depending on specific applications, a MAC protocol can incorporate TDMA or/and CSMA with other techniques to meet different performance requirements.

There is no generic best MAC protocol; the proper choice depends on the application, the expected load patterns, the expected deployment (sparse versus dense sensor networks), and the specifics of the underlying hardware's energy-consumption behaviour, for example, the relative costs of transmitting, receiving, switching between modes, wakeup times, and wakeup energy from sleep mode as well as the specific computation costs for executing the MAC protocol.

2.9 Cross-layer approaches

The main drawback of the WSN protocols is that they follow traditional layered protocol architecture [Akyildiz et al., 2002a; Akyildiz and Vuran, 2010]. Although, they may achieve very high performance in terms of all the metrics related to each of the individual layers, they are not jointly optimised to maximize the overall network performance, while minimizing the energy expenditure. Transmission power is an important metric for measuring energy consumption. A change of the power level may result in a change of route selection, and accordingly change the paths and hence affects the network connectivity. Also, energy consumption for data transmission, data processing and reception must be take into account, emphasising the need of cross-layer approaches across all layers of the protocol stack. Cross-layer interaction is considered where the traditional layered architecture is preserved while each layer is informed about the state of other layers [Akyildiz et al., 2002a]. In cross-layer protocols, the functionalities of different layers in the communication protocol stack are considered in a joint fashion with the objective to optimize some performance metric of interest, and not handled separately like in a traditional network, for example, minimize the overall energy consumption

of the communication protocol. However, the internal architecture of each layer still stays intact. Therefore, power control is a cross-layer design problem, which affects all layers of the protocol stack, from the physical layer to the transport layer, and thus has a great impact on several key performance metrics, including throughput, delay, and energy consumption [Zheng and Jamalipour, 2009].

Although, the communication protocols that focus on pairwise cross-layer interaction provide beneficial insights, the ultimate goal is the development of more general cross-layer module that provides the necessary functionalities of a WSN without the inabilities of the layered protocol stack, and yield much more optimised solutions for resource-constrained devices. Typical interlayer effects in WSNs is that between the physical and the routing layers, MAC and routing layers, MAC and application layers etc. Therefore, cross-layer optimization is highly desirable and stands as the most promising alternative to inefficient traditional layered protocol architectures, incorporating transport, routing, MAC and physical layer functionalities into a single cross-layer module.

In WSNs, the unreliability of the links and the limitations of all resources bring considerable complications to routing. Even in the presence of static nodes, the channel conditions vary because of multipath fading effects due to the motion of people or objects in the environment, which modify the patterns of radio wave reflections. Also, sensing nodes are typically battery-powered, and ongoing maintenance may not be possible: the progressive reduction of the available energy needs to be factored in. The quality of the links and the remaining energy in the nodes are the primary factors that shape the network graph; link quality may be measured directly by most radios, whereas residual energy is related to the node battery voltage, which may be measured and fed into the micro-controller. These quantities may be used to form a cost function for the selection of the most efficient route. Moreover, the presence of a time constraint requires the network to favour routes over a short number of hops(a.k.a.the long-hop approach, in the sense that a small number of long hops is used) in order to minimize delay. Hop number information may be incorporated into the cost function to bias route selection toward minimum-delay routes. Thus, a cross-layer cost function is obtained, which includes raw hardware information (remaining energy), physical layer data (channel quality), and a routing layer metric (number of hops).

Given the broadcast nature of the wireless medium, sensor nodes in WSNs become a potential target for attacks [Chen et al., 2010]. Malicious nodes can join the network and eaves drop in order to damage transmissions or steal information. Some attacks have been identified in the literature and also some attack detection and protection techniques have been proposed.

The great number of cross-layer approaches that address the challenges presented by the new applications of WSNs proves that there is still need for further optimization of these networks, and that cross-layering is efficient to accomplish that. At the application layer, QoS provisioning is the main challenge. For the MAC layer, duty-cycles scheduling has proven to save energy with different designs for each medium access method. And at the physical layer, some effects of the use of multilevel modulation and the consideration of parameters measured at this layer,e.g.,CSI and REI, by other layers has been considered in literature. Furthermore, some of the standards considered in cross-layer approaches have also been considered. In the future, from the application layer, it has been seen that strict QoS parameters may need to be respected depending on the running application. However, as it can be counter productive for the WSN lifetime, further investigation needs to be carried out to verify the behaviour of the used multimedia coder (MPEG, H.263, H.264) on the sensors energy consumption. Generally, the work that considers these QoS requirements and coders only consider packet error rates at the physical level.Thus,as more complex channel models are available in research work that does not consider the application layer,these proposals could be combined to result in a more detailed design that considers the effects of the physical characteristics on the application. Also, none of the discussed work involving the application layer has been deployed, and thus only analytical and simulation results are available. This is an interesting challenge to be addressed in the future. Moreover, routing comprises the most challenging set of issues on WSNs. There is no consensus on the best approach for routing since some proposals consider clustered networks, others consider flat routing (no hierarchy), and also some consider centralized routing. The number of sensors in the WSN can clearly affect the performance of each approach, requiring scalability tests that have not been done in the literature. Furthermore, sensors mobility are rarely considered. The high complexity of the proposed routing protocols for WSNs is increased by consider-

ing mobility, demanding a great amount of effort from researchers to fill this gap. Also concerning the network layer, MIMO effects on routing protocols have been only superficially verified, and spatial diversity has not been explored yet. Besides, cross-layer proposals do not agree on the most efficient medium access method for WSNs. Several MAC protocols have been created based on TDMA and CSMA, but the literature lacks of the comparison between them. In addition, boundaries between uplink and downlink phases, optimal frame sizes, and synchronization methods are still open issues. Finally, to properly address the efficiency of the different modulation and transmission technologies considered in cross-layer design, an extensive comparison between them is needed. Clearly, as they are closely related to the channel properties and energy consumption profile, they directly affect the parameters at the upper layers. Hence, there is still much to be done in order to achieve a comprehensive cross-layer design that addresses the issues at every layer of the stack in an energy-efficient manner, but also considering the application requirements.

2.9.1 Energy-aware methods

Sensor nodes are almost invariably constrained in energy resources and radio channel transmission bandwidth; these constraints, in conjunction with a typical deployment of large number of sensor nodes, have posed a plethora of challenges to the design and management of WSNs. These challenges necessitate energy awareness at all layers of a communications protocol stack. Researchers have developed many new protocols specifically designed for WSNs, where energy awareness is an essential consideration; focus has been given to the routing protocols, since they might differ from traditional networks (depending on the application and network architecture). A detailed survey is presented on the energy aware methods in WSNs [Akkaya and Younis, 2005].

2.10 Need for a plug-in

Currently, modelling is used to specify a WSN at different levels of abstraction (hardware, application, communication protocols, etc.) with the recurrent goals

of code generation, communication overhead analysis, energy consumption. However, WSNs lead to many challenges such as abstraction, separation of concerns and reuse [Stankovic, 2004]. When current practices in WSNs are considered, it is quite evident, the lack of engineering methods and techniques to manage these challenges. Recently, the need of abstracting an implementation view into an architectural design is getting more realized. A clear separation of concerns is needed as the hardware and software aspects are locked and tied down only to a specific type of nodes, hampering the possibility of reuse across projects and organizations. This means that exploiting the right level of abstraction, and keeping explicit (and separated) software and hardware architectural details will surely ease developers job.

Separation of concern is limited as the hardware and software components are locked and tied down to specific types of nodes, hampering the possibility to reuse components across projects and organizations. Moreover, while the focus is mostly on software components and hardware, there is still a missing piece from the WSN modelling puzzle: the physical environment where the WSN application will be deployed. Since the physical environment plays a fundamental role especially when the energy consumption of WSNs is considered, lack of an explicit representation of the physical environment is an important limitation of existing approaches. Under this perspective, approaches abstracting implementation details from the underlying hardware and physical infrastructure are strongly advised [Blumenthal et al., 2003; Mottola and Picco, 2011]. Some initial effort has been conducted for architecting WSNs [Hill, 2003; Losilla et al., 2007], however, they could only meet the expectations partially. Currently, modelling is used to specify a WSN at different levels of abstraction (hardware, application, communication protocols, etc.) with the recurrent goals of code generation, communication overhead analysis, energy consumption. The authors addressed energy-aware system design of Wireless Sensor Networks in [Gotzhein et al., 2009]. Energy mode signalling and energy scheduling of nodes within a WSN are represented as SDL models and then analysed. A framework for modelling, simulation and code generation of WSNs is presented in [Mozumdar et al., 2008]. The framework is based on Simulink, State-flow and Embedded Coder, allowing the engineers to simulate and automatically generate the code, with energy as one of the major issues. A model-driven

approach has been proposed in [Shimizu et al., 2011], which enables a low-cost prototyping and optimization of WSN applications. In this work, a set of modelling languages is the starting point for code generation and performance (with energy consumption) analysis.

The existing evaluation tools (TOSSIM [Levis et al., 2003], Atemu [Polley et al., 2004], Avrora [Titzer and et al., 2005], Shawn [201, 2012], AlgoSensim [Jacques and Marculescu, 2011], and Sinalgo [201, 2011d], OPNET [201, 2011a] and Qualnet [201, 2011c]) use rather simple radio/channel models [Kotz et al., 2004]. Also, the simulators are still platform specific and moderately scalable, making them unsuitable for protocol/algorithm design and testing. The major power consumption of the node is based on the time the radio is on, either transmitting, receiving or listening, and how long the radio stays in each of the states. Hence, it is also of significant importance to consider the energy consumed for listening as well, for performance evaluation. Furthermore the environmental details and especially the effects of path loss, effect of collisions, impact of security mechanisms and clustering techniques have not been considered in any of the existing evaluation tools.

The existing simulation tools for WSN are presented in the table below. Extensive details on these simulation tools is given in chapter 5.

Table 2.1: Existing simulation tools in WSN

Instruction Level Simulators	Algorithm Level Simulators	Packet Level Simulators
TOSSIM	Shawn	OPNET
Atemu	AlgoSensim	Qualnet
Avrora	Sinalgo	NS2, GloMoSim

2.11 Need for WSN Architecture

Despite the ever increasing usage of WSNs in modern applications, their development is still plagued by the following issues: (i) development is still performed directly on the top of the operating systems and relies greatly on individuals hard-earned programming skills across all levels of the communication stack (e.g., ap-

plication, routing, data link levels, etc.) [Mottola and Picco, 2011]; (ii) challenging extra-functional requirements such as performance, security, energy consumption, with poor support for early testing, debugging, and simulation of the WSN in an integrated fashion must be addressed by WSN engineers [Imran et al., 2010]; (iii) in order to achieve the desired level of efficiency, the software of a WSN application is tied to specific hardware platforms, thus hampering the reuse of source code and software components across different projects or organizations [Mottola and Picco, 2011]; (iv) due to the intrinsic multidisciplinary nature of the WSN problem space, WSN engineers must continuously collaborate with a high number of system stakeholders (e.g., WSN users, application domain experts, hardware designers, and software developers) with different background and training [Romer and Mattern, 2004a].

In current practice, programmers do not only face functional requirements but also challenging non-functional requirements such as lifetime, performance, and security. Besides the need of programming abstraction, it is well-accepted the need of abstracting an implementation view into an architectural design. As remarked in [Picco, 2010], "end users require high-level abstractions that simplify the configuration of the WSN at large, possibly allowing one to define its software architecture based on pre-canned components". Abstraction is fundamental for future WSN development, as sensors and WSNs in general are becoming important components in pervasive, mobile systems, with new types of stakeholders (e.g., mobile systems engineers, developers) with reduced domain specific technical skills.

As possible solution to the above mentioned issues, the WSN community is becoming aware of the need of using software engineering approaches in order to support the design, analysis, simulation and implementation of WSNs [Picco, 2010; Willig, 2006b].

In order to simplify the design and configuration of the WSN at large, and abstract from technical low-level details, a number of Model-Driven Engineering (MDE) approaches for WSN engineering have been proposed. Currently, these approaches are used to specify a WSN at different levels of abstraction (hardware, application, communication protocols, etc.) with the recurrent goals of code generation, communication overhead analysis and energy consumption. Many ap-

proaches intend to use Domain-specific Modelling Languages (DSML) for representing WSNs from different viewpoints. For example, the authors in [Vicente-Chicote et al., 2007] proposed modelling languages with concepts such as node group, region, resource, wireless link; whereas, the authors in [Imran et al., 2010] proposed a set of languages spanning from application-level actions (e.g., sense, send message, store data) to hardware specifications (e.g., processor, sensing devices, radio transceivers), and so on. Other approaches, such as those proposed by the authors in [Fuchs and German; Mozumdar et al., 2008], are based on generic modelling languages; mainly, they use extensions of UML and Simulink for representing a WSN. For what concerns the physical environment of the WSN is that, the majority of approaches in the literature does not allow designers to specify the physical deployment of the WSN nodes. Among those that support this feature in some way, there is great variability. There are some that supports an explicit definition of the physical environment , others that allow the designers to define physical quantities (e.g., in [Ben Maissa et al., 2012] engineers can define models of the evolution of each physical quantity in a given scenario), and so on. However, all these approaches do not provide any intuitive and abstract means to easily define the deployment environment of the WSN.

Hence, the need for an approach which considers the collaborative nature of WSNs along with its correlation characteristics and various issues from physical layer to application layer together as entities to enable a framework. Also, there is a need to compare work on existing work on collaborative architectures/methods of WSNs. A clear separation of concerns is also needed as the hardware and software aspects are locked and tied down only to a specific type of nodes, hampering the possibility of reuse across projects and organizations. Along with this, a realistic WSN life time estimation and performance evaluation should be possible, in an attempt to improve performance maximizing the lifetime of the network.

Chapter 3

Modelling WSNs

3.1 Introduction

Wireless Sensor Networks rely on cooperative effort of the densely deployed sensor nodes to gather information from the habitat [Akyildiz et al., 2002b; Bouabdallah et al., 2009; Chiasserini and Garetto, 2004] typically to achieve either environmental monitoring or target tracking and sensing. Depending on the area of application, information monitoring and reporting may further be classified as continuous, periodic, or event-based (driven) [Bouabdallah et al., 2009; Wang et al., 2012]. An example may be temperature monitoring where the first case involves reading and reporting periodically irrespective of the changes involved, the second scenario may be where only variations from previous readings are reported and finally, the case where a report is sent only when a specific temperature is reached. In all these cases, data arrival delay is clearly determined by the nature of application and the chosen monitoring scheme. Apart from the common challenges of WSNs including energy consumption, network connectivity, data aggregation, computation power, limited sensor node memory, the end to end delay of transmitted packets remains a serious concern in relation to Quality of Service (QoS) provision [Akyildiz et al., 2002b]. In [Wang et al., 2012], cross layer analysis of the end to end delay distribution in WSNs was studied and the results show that inter-arrival time (time between two consecutive arrivals) mostly follows exponential distribution except for low periodic traffic. There are many studies which consider exponential ar-

rivals to sensor nodes [Omondi et al., 2013b; Zhang and Li, 2011, 2012]. However, in other quarters there has been mixed opinions on the appropriate distribution for modelling inter arrival delay of WSN data packets [Chiasserini and Garetto, 2006, 2004; Wang et al., 2012]. In other works, there has been mixed opinions on the appropriate distribution for modelling inter-arrival time of WSN data packets [Chiasserini and Garetto, 2004]. This strongly indicates the need for a study to identify acceptable types of distributions for inter-arrival times used in modelling WSNs. Characterization of the end-to-end delay distribution is fundamental for real-time communication applications with probabilistic QoS guarantees. Indeed, the cumulative distribution function (CDF) of the delay for a given deadline can be used as a probabilistic metric for reliability and timeliness[Wang et al., 2012]. Researchers have also continued to develop algorithms and protocols to address some of the challenges like balancing cluster energy consumption in clustered WSNs as well as path loss effects [Ever et al., 2012].

In this chapter, an investigation is carried out to establish the most appropriate distribution for the inter-arrival times at Cluster Heads (CH) and relay nodes. The process is started by identifying and characterizing various applications and determining suitable data delivery models depending on application requirements. Simulation results are presented and analysed in detail to characterize end to end delay between arriving data packets. The effects caused by medium access control (MAC) protocol properties are also analysed by experimenting with well known MAC protocols. Most existing WSN simulators assume that exponential distribution is valid for characterising arrivals of data packets at nodes within the WSN.To the best of the our knowledge, this is the first work that provides statistical substantiation of the results along with probabilistic analysis of arrival distributions at the CH or relay nodes in WSNs. There exists plenty of literature [Chiasserini and Garetto, 2004; Wang et al., 2012] that the probability of an event occurring at any time is governed by a Poisson‘s process, and the inter-arrival times are Exponentially distributed. This is the main assumptions of the existing studies and in fact in this study the validity of such assumptions is checked.

Kolmogorov-Smirnov (K-S) test statistics are used to decide whether a certain type of distribution function assumption is appropriate for inter-arrival time distribution. The rest of the chapter is organised as follows: Related work in this

area is summarized in Section 3.2. Section 3.3 discusses the data delivery models that are characterised based on the application requirement along with the related performance aspects. A detailed description of the communication paradigm considered in this work is presented in Section 3.4, followed by the system's detailed discussions on inter-arrival distributions along with various aspects of the case studies like the effects of MAC, data rates and application types are provided in Section 3.5. Simulation results for inter-arrival distributions at the CH are presented along with their equivalence using statistical studies and further probabilistic analysis are presented in Section 3.6. Finally, Section 3.7 concludes the chapter with detailed explanation about various distributions in Appendices 9.

3.2 Performance Modelling of WSNs: Related Work

Performance modelling and analysis continues to be of great importance in supporting research as well as in the design, development and optimization of WSN and their applications. The current trend towards the use of WSNs for sensing and control now has the potential for significant advances, not only in science and engineering, but also, on a broad range of applications. This brings the need for performance modelling for the optimization of deployment of WSNs. However, the special design, characteristics of sensors and their applications separate them from the traditional networks. These characteristics pose great challenges for the architecture, protocol design, performance modelling and their implementation. It is essential to consider energy efficiency of WSNs because of their limited energy sources (most of the times batteries). In order to minimise the energy consumption, one of the effective techniques is to place sensors in sleep mode during the idle period [Singh and Raghavendra, 1998]. In [Schurgers et al., 2002; Ye et al., 2002; Zheng et al., 2003], a wake-up scheduling scheme at the MAC layer is proposed, which wakes up the sleeping nodes when there is a need to transmit or receive, thus avoiding a degradation in network connectivity or quality of service provisioning.

Characterising delay in distributed systems has been considered in various contexts. However, it can be observed that accurately characterizing end-to-end delay at the CH is still an open problem. Considerable amount of research on sensor networks reported recently has been ranging from network capacity and signal

processing techniques, to topology management, algorithms for traffic routing and channel access control. The model presented in [Chiasserini and Garetto, 2004] is used to investigate system performance in terms of energy consumption, network capacity, delay in data delivery along with the trade-off's that exist between performance metrics and sensor dynamics in active/sleep modes. A Markov model is presented for WSNs, where the nodes may enter into sleep mode. Through standard Markovian techniques, a system model representing the behaviour of a single sensor has been constructed along with the dynamics of the entire network, and the channel contention among interfering sensors. The proposed solution of the system model is then obtained by means of a Fixed Point Approximation (FPA) procedure, and the model has been validated via simulation.

Due to hardware constraints for energy efficiency, optimizing node packet buffer and maximizing the performance is necessary to improve the Quality of Service(QoS) for transmission in WSNs. In [Qiu et al., 2011], a packet buffer evaluation method using Queuing Network Models is proposed where, the blocking probabilities and system performance indicators of each node are calculated using an approximate iterative algorithm. The model considered focuses on a single server model in WSNs and the method used to calculate packet buffer capacity for nodes also indicate that the sink node requires higher performance, when compared to the other nodes in the network. The Markov model of the sensor sleep/active dynamics is presented in [Mini et al., 2002], that predicts the sensor energy consumption by acquiring this information for each sensor, while a central controller constructs the network energy map representing the energy reserves available in various parts of the system. Only a single node is represented by a Markov chain, while the network energy status is derived with the help of simulation studies.

With regard to analytical studies, results on the capacity of large stationary ad-hoc networks are presented in [Gupta and Kumar, 2000]. Two network scenarios were considered; one including arbitrarily located nodes and traffic patterns, while the other one with randomly located nodes and traffic patterns. An analytical approach on network coverage and connectivity of sensor grids is presented in [Shakkottai et al., 2003]. The sensors are considered unreliable and fail with a certain probability leading to random grid networks. Results on coverage and connectivity are derived as functions of key parameters such as the number of

nodes and their transmission radius.

Several approaches based on simulations and experiments, have been proposed for performance evaluation of IEEE 802.15.4 networks [Ferrari et al., 2007]. In [Bianchi, 2006], an analytical framework based on a Markov chain characterization of the MAC protocol is proposed for IEEE 802.11 networks in saturation conditions. Based on this pioneering work, several approaches have been proposed for the characterization of the MAC performance in IEEE 802.15.4 networks with a star topology. In this work, a scenario with acknowledgement (ACK) messages is considered and an evaulation of the network performance in both saturation and non-saturation regimes is presented, while trying to characterize the conditions under which the network enters the saturation region [Misic et al., 2006]. A simple Markov chain theoretical model to characterize the sensors as well as the channel status is proposed in [Ramachandran et al., 2007]. The models shows good agreement with ns-2 based simulations. This model allows to investigate throughput and energy consumption metrics within WSNs. In [Martalò et al.], an extended framework of the one proposed by [Ramachandran et al., 2007] is presented for a 2-hop network scenario, i.e., networks where sensors communicate with the coordinator through an intermediate relay node, which forwards data packets from the sources (the sensors) towards the destination (the coordinator). Similar works have been presented in [Misic and Udayshankar; Misic et al., 2005], emphasising the use of a relay for interconnecting two different clusters in IEEE 802.15.4 networks and analysing the performance through a queueing theoretical analysis. However, the proposed scenario models the (simpler) cases where the relay does not content the medium access to the sensors. Hence, it is observed that accurately characterizing arrivals at the cluster head in WSNs is still an open problem. Although it is quite difficult to analyse each possible application in WSNs, it is sufficient to analyse each class of application classified by data delivery models, as most of these applications in each class have common requirements on the network [Emary and Ramakrishnan, 2013]. A well established simulation tool Castalia which provides realistic node behaviour, wireless channel and radio models, and enables to mimic and analyse the real life scenarios for various types of applications is employed in this study.

3.3 Characterising Data Delivery Models

Although it is quite difficult to analyse each possible application in WSNs, it is sufficient to analyse each class of application classified by data-delivery models, as most applications in each class have common requirements on the network [Emary and Ramakrishnan, 2013]. From the point of view of network QoS, the network is concerned with how to transmit the sensed data from the sensor field to the sink node, fulfilling the corresponding required QoS. The factors that characterize the application requirement are presented in Table 3.1.

The practical realization of the current WSN applications depends on the energy-efficient, real-time and reliable communication capabilities of WSN. WSNs have distinct traffic characteristics. The primary traffic is generally a many to one type communication, i.e., from the sensor nodes to the base station, in the upstream direction. Upstream traffic delivery can be classified as: continuous, event driven, query driven and hybrid-based data delivery models. Depending on their specific applications, these data delivery models have different quality of service and reliability requirements [Tilak et al., 2002]. Traffic model greatly influences protocol design and affects performance. The four models and the related performance aspects are discussed below.

Table 3.1: Application Requirements of Data-Delivery Models

Factor	Event-Driven	Query-Driven	Continuous	Hybrid
Interactivity	✓	✓	✗	✓
End-to-End Performance	✗	✗	✗	✗
Delay Tolerance	✗	Query-specific	✓	✗
Criticality	✓	✓	✓	✓

- In continuous delivery model, each sensor reports regularly, perhaps continuously, or periodically, to the sink at a pre-specified rate. Some networks apply a hybrid model using a combination of continuous, event-driven and query-driven data delivery. Time based medium access control protocols can be used to achieve significant energy savings in case of continuous data delivery models.

- Most event-driven applications in WSNs are interactive, delay intolerant (real-time), mission critical and non-end-to-end applications. When an event occurs, the sensor node begins to report the event, and possibly an associated value, to the sink. The application needs to receive the desired data reliably and as quickly as possible. The query-driven data delivery model is very similar to the event-driven model, except that the data are pulled by the sink where as in event-driven models, the data are pushed to the sink. The application in most of the cases may not be an end to end one, i.e., one end of the application is the sink, where as in the other end, a group of sensor nodes within the area that are influenced by the event. Also, the traffic generated by a single sensor node may be of a very low intensity, however, more random and unforeseeable bursty traffic may be generated by a set of sensors due to the event occurring. Therefore, the routing efficiency for this delivery model is heavily dependent on the frequency of occurrence of the events. CSMA medium access arbitration is a good fit for event-based data delivery models since the data are generated sporadically.

- Similar to event-driven applications, most query-driven applications in WSNs are also interactive, mission critical, query-specific, delay tolerant and non-end-to-end applications. Queries can be sent on demand in order to save energy. Sometimes, the base station may be interested in a specific piece of information that has already been collected by the sensor nodes. The sensor only reports the observed data in response to an explicit request from the user. In this delivery model, the sink broadcasts the query message, a path is constructed automatically when the query arrives at the sensor nodes, and the sensor nodes report their findings according to the request in the query message. Query-driven systems store gathered information locally and communicate it on request. This type of sensor network can be useful in logistics or home applications, but is not very common in applications of environmental Monitoring.

- Some networks apply a hybrid model using a combination of continuous, event-driven and query-driven data delivery, as the types of sensors and the data they sense may be very diverse. For example, data may be reported

continuously by some nodes, and the sink may need to query information from other sensor nodes.

3.4 Our Sensor Communication Paradigm

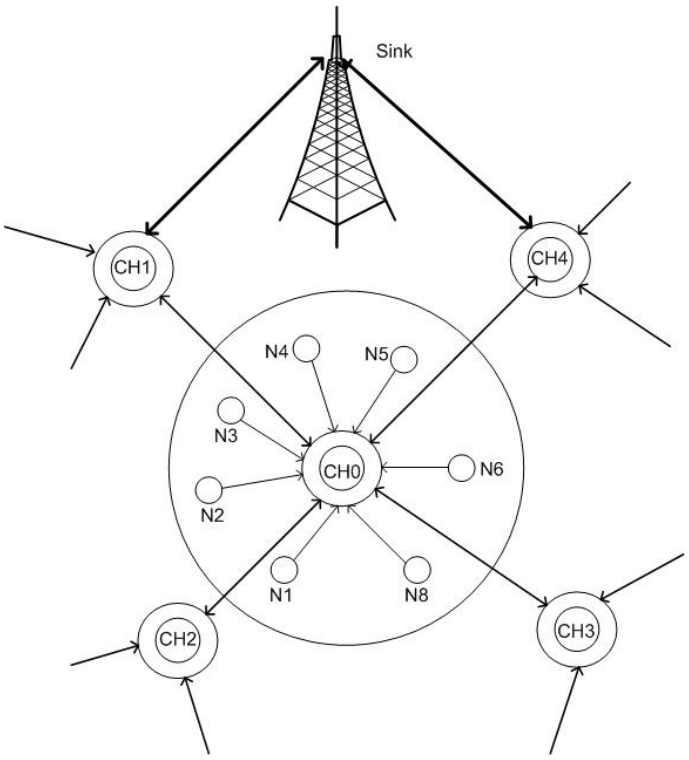

Figure 3.1: Network topology of the reference scenario

A system of Wireless Sensor Network with identical sensor nodes deployed in a cluster tree topology is considered. The sensor nodes used are assumed to self-configure during initial deployment and remain stationary thereafter. All the nodes in a cluster and adjacent CHs are considered directly connected to the CH. The primary focus is to study the inter-arrival distribution of packets at the CH. The total arriving data packets at the CH at any given time is therefore equal to the sum of all the independent arrivals from the cluster nodes and arrivals from adjacent CHs forwarding their data to the sink. For this case continuous monitoring of event driven systems are considered.

In this set up all nodes are considered to be equipped with an omnidirectional antenna and they also have a common maximum radio range r within which they are able sense event occurrences and also transmit information to the CH based on the 802.15.4/Zigbee standards. The topology of interest is shown in Figure 3.1. For simplicity, all sensor nodes are shown connected directly to the CH0 in Figure 3.1. CH0 can forward data to the sink either through CH1 or CH4, whereas CH2 and CH3 forwards their packets to the sink passing through CH0. It is also shown that nodes N1 to N7 are directly connected to the CH0. For each of the cases considered in 3.14, the number of nodes in each application is varied from 10 (10 nodes in each cluster connected to the cluster head) up to 40 nodes (40 cluster nodes connected to each cluster head). One cluster head is considered with the cluster nodes forwarding the data to it, while other cluster heads are forwarding their cluster head as an aggregate towards the sink. The case studies considered is just an example of a typical scenario considered. Clustering protocols are compared to find the best one available amongst LEAH, unequal LEACH, HEED and UHEED. UHEED is used to choose the CH in each of the application. More details on these protocols are presented in Chapter6. The typical functions provided by the different types of sensors considered for the three applications mentioned in 3.14 are temperature sensor, pressure sensor, inclinometers, vibration sensors, fluid property analysers, traffic sensors, motion sensors, ECG, EEG, etc.

Each sensor node is able to independently monitor its habitat and organise the information sensed into fixed data units storable at the sensor buffer before finally forwarding to the CH. The buffers, both at the sensor nodes and at the CH are assumed to have infinite capacity and are follows First in First out (FIFO) queuing discipline. The Cluster Head is only able to receive or transmit at one go within the assigned time slots of unit duration. Once Information sensed and aggregated at the nodes are forwarded to the CH, it finalizes cluster aggregation and transmits all the information to the sink either directly or through other intermediary CHs. It is assumed that at least one path always exists towards the sink [Chiasserini and Garetto, 2004].

In order to analyse the inter-arrival distribution at the CH, use of Castalia is employed. For each experiment, packet arrival rate and number of nodes is set at desired values. Desired MAC properties; T-MAC, CSMA, and no MAC

are then considered for each experiment. The generated inter-arrival distribution time results are then further analysed using statistical tools to identify the actual distribution pattern.

3.5 Detailed Analysis of Case study and Simulations

3.5.1 Inter-Arrival Distributions

Providing QoS guarantees in terms of delay, jitter, and throughput has been the main focus of researchers, as the connectivity between different domains improves. End-to-end QoS guarantees are complicated by the inherent differences in the nature of the wireless media. Therefore, providing QoS guarantees in a network, in general, requires sophisticated traffic management and admission control procedures. This requirement is even more important in networks of low-power, low-data-rate sensor nodes, where network resources are scarce and dynamic. Considering the non-deterministic nature of communication due to wireless channel errors and traffic characteristics, probabilistic analysis of network performance is crucial to provide QoS guarantees.

One of the most important metric of QoS is the probability distribution of inter-arrival times of packets in WSNs. In order to characterise the distribution of packet inter-arrival times, the number of arrivals is considered from the numerical results provided by Castalia. In a typical cluster network, the inter-arrival time is characterised by the following: the resulting job arrivals at the CHs is a collection of jobs from locally generated packets and relay packets from other neighbouring CHs. Locally generated packets consists of the sensed information by the CH itself and from other cluster nodes in the clusters. The jobs are independent and are identically distributed random variables with an arrival rate λ. We carry out investigation to establish the most appropriate distributions suitable for modelling the inter-arrival distributions of these packets at the CH. The inter-arrival time of the packets received by the CH depends on the application requirements, with which the sensor data are generated. The generated traffic mainly depends on the

physical phenomenon of interest and the type of application, while the relay traffic depends on the network parameters. For evaluation purposes, a clustered network is considered where the inter-arrival distribution is found for the CH under the contention from the cluster nodes. The distribution of the inter-arrival time of the packets is recorded at the CH. Each of the nodes are inter-related according to the traffic constraints. Each cluster node transmits its generated packets to the corresponding CH, where the CH aggregates the packets received from its cluster nodes, along with its own generated packets and relay packets from other neighbouring CHs and forwards them to the next CH on route or directly to the sink. In other words, the sum of the incoming relay traffic rate at each CH is equal to the transmitted traffic rate from each of the cluster node.

3.5.2 Event-driven and Continuous-monitoring Applications

The arrival time of the generated packets from each sensor node send to the CH depends on the application requirements, from which the sensor data are accordingly generated. Depending on the type of application, i.e., in case of event-based applications, the sensor node begins to report the event and possibly an associated value to the CH or to the sink (if the node is a CH itself), when an event occurs. In such cases, the data generated are often sporadic. Considering such physical events, e.g., fire alarm system, temperature sensing systems etc., the event being monitored do not occur very frequently, i.e., occuring at irregular intervals in time. Extensive work has been already carried out in estimating the distribution of inter-arrival time of the packets at each node, considering physical events that do not occur very often. In [Wang et al., 2012], it was shown that the probability of any event occurring at any time is governed by a Poisson's process, and the inter-arrival times are exponentially distributed. Query-driven applications are also very similar to event-driven applications in terms of arrival time of the generated packets from each sensor node to the CH. This is because they also depend on the application requirements.

In applications involving the source sensors sending their sensed data continuously to the sink, for example, in a temperature-sensing systems, the sensors send their data to the cluster head/sink in a continuous manner throughout the time,

at a specified rate. The deployment at Great Duck Island [Mainwaring et al., 2002] is an example of a continuous monitoring network, where the nodes are capturing the movement of Petrels once every 5 to 10 minutes. The class of continuous data delivery model can be further classified, depending on the data rate of operation. Although, WSNs are usually considered as very low data rate networks, there is a great potential to utilize the benefits of WSNs for high data rate and low delay demanding applications, such as media streaming and critical control. Examples of low data rate sensors include temperature, humidity, and peak strain captured passively whereas, examples of high data rate sensors include strain, acceleration, and vibration sensors.

3.5.3 Effects of MAC

Channel contention plays an important role in causing additional delay, queuing delay and wireless channel errors at the CH due to the job arrivals from cluster nodes and forwarded data from other CHs. The MAC layer is responsible for scheduling and allocation of the shared wireless channel which eventually determines the link level QoS parameters, namely MAC delay. MAC protocols provide the greatest influence over communication mechanisms and provide the most direct influence over utilization of the transceiver, as transceiver that constantly senses the channel will quickly deplete the sensor node energy resources and shorten the network lifetime to unacceptable levels. The main design goal of a typical MAC protocols is to provide high throughput and QoS. On the other hand, wireless sensor MAC protocol gives higher priority to minimize the energy consumption rather than the QoS requirements. Hence, characterization of inter-arrival distribution is fundamental and can be used as a probabilistic metric to estimate the QoS in WSNs. Channel contention is a serious problem in WSNs resulting in collisions, re-transmissions, energy depletion, and ultimately loss of event reports. MAC protocols employ a back-off algorithm to resolve contention among nodes to acquire channel access. Most common contention-based MAC protocols can be employed such as CSMA or T-MAC for transmissions to keep the energy consumption low, reducing the amount of energy wasted on idle listening, in which nodes wait for potentially incoming messages, while still maintaining a reasonable throughput

[van Dam and Langendoen, 2003].

Majority of the WSN MAC protocols are contention-based, wherein the contention window size setting involves an important trade-off between the collision probability and idle listening durations in contentions where both are aimed to be lowered for efficient network operation. Sensor network MAC protocols often trade performance characteristics, such as throughput and latency, for a decrease in energy consumption to lengthen a sensor node's lifetime. The key challenge of supporting real-time data transmission in CSMA-based model is the non-deterministic nature of delay for a successful transmission of a data packet. CSMA/CA is a contention-based technique where the node needs to sense whether the channel is idle before it can transmit a packet. If the channel is not idle at that time, the node needs to wait for a certain period of time before it can sense the channel again. This scheme makes the delay time for a successful transmission non-deterministic. If there is a duty cycle in place (for example, like in T-MAC) then whenever the node back-off's, it also goes to sleep. Every node periodically wakes up to communicate with its neighbours, and then goes to sleep again until the next frame. Mean while, new messages are queued. Nodes communicate with each other using a Request-To-Send (RTS), Clear-To-Send (CTS), Data, Acknowledgement (ACK) scheme, which provides both collision avoidance and reliable transmission. A node will keep listening and potentially transmitting, as long as it is in active period. An active period ends when no activation event occurs for a predefined time. A node will sleep if it is not in an active period, consequently, the predefined time determines the minimal amount of idle listening per frame. The described time out scheme moves all communication to a burst at the beginning of the frame. Since messages between active times must be buffered, the buffer capacity determines an upper bound on the maximum frame time. When it is time to wake up there is an extra delay before the node can sense the channel. The nodes (excluding the sink) turn off their radio periodically to save energy. When any node has a packet to send, it starts to repeatedly transmit request to send (RTS) beacon packets based on CSMA/CA manner, i.e. through carrier sense and random back-off manner, and therefore causing delays. This extra delay slightly increases the probability of collisions, but on the other hand, a node can save considerable energy (especially in heavy traffic where it is backing off often) [van Dam and Langendoen, 2003].

Some parts of the delay can be governed by equations, such as the transmission time of a packet by the radio which obviously depends on the data rate of the radio and the size of the packet. But most of the delay happens because of the MAC protocol. An example could be that the MAC layer is waiting for the channel to be clear or waiting for the active period to commence. Since retransmissions involve the MAC each time, then most of the delay of a retransmission packet will be due to the MAC as well. Hence, buffering the packet for retransmission, will cause unnecessary delay, affecting the performance of the network. The delay distribution models presented in the literature do not consider the uncertainties due to random back-off's because of the MAC protocols. Therefore, it is quite an important task to characterise the inter-arrival distributions at the CH especially when considering the affects of MAC protocols, causing delays.

3.5.4 Case Study and Simulation Parameters

Most of the applications in WSNs have common requirements on the network. Therefore, in order to analyse possible applications, a case study, which is a typical scenario is considered resembling an application area where the cluster nodes send the sensed data at a constant rate to the CH. Importantly, in order to characterise the distribution of inter-arrival time of packets, case studies based on typical scenarios are considered. The real contribution here is to provide statistical substantiation of the results along with probabilistic analysis of statistical distributions of the arrival distributions at the CH and relay nodes in WSNs.

Simulation results are obtained with simulation package Castalia, the WSN framework of OMNET++. It is mainly used for initial testing of protocols and/or algorithms with realistic node behaviour, wireless channel and radio models. The OMNeT++ platform is an extensible, modular, component-based C++ simulation library and framework, primarily for building network simulators. Castalia is highly tunable, features an accurate radio model based on the work of the authors in [Zuniga and Krishnamachari, 2009]. It also features physical process model, considering clock drift, sensor energy consumption, CPU energy consumption, sensor bias etc. Specific details related to unpredictability of the wireless channel, energy spent in transmission/receiving packets, performance degradation experienced by

duty cycles, collisions are well established in Castalia [Newport et al., 2004].

In order to characterise the distribution of packet inter-arrival times at the CH, a typical scenario in WSN applications considering constant transmissions from nodes to CH, having cluster networks of various sizes (from 10 nodes to 40 nodes) are considered. The following parameters are used throughout the simulations, unless otherwise stated. A CC2420 chip, compatible with 802.15.4, is used to provide wireless communication, operating at 2.4 GHz and providing a data rate of 250 kbps. For T-MAC and CSMA, the internal MAC buffer size in packets is 32. The packet size is considered to be 105 bytes [LatrÃľ et al., 2005]. The simulation scenarios are chosen from the examples of prototyped applications for WSNs presented in the literature [Kuorilehto et al., 2005]. Although it is quite difficult to analyse each possible application in WSNs, it is sufficient to analyse each class of application classified by data delivery models, as most of these applications have common requirements [Emary and Ramakrishnan, 2013]. R free software environment which provides wide variety of statistical and graphical tools is is used for K-S test and statistical evaluation.

3.6 Numerical Results

In table 3.14 below, we report the results of finding theoretical distributions to the empirical arrival distributions of simulated data series at the CH and intermediary routing nodes. The first column presents the number of observations in the simulated series. The second column displays estimated Maximum Likelihood parameters of empirical distributions [1]

The well-known theoretical distributions corresponding properly to the empiri-

[1] When the joint density for a set of variables is viewed as a function of the parameters alone, that function is called a *Likelihood function*. Hence the Likelihood function, $L(\theta)$, is defined as $L(\theta) = f_\theta(x)$. Here $\log f_\theta(x)$ is a scalar function of a k-dimensional variable θ and $x = (x_1, x_2, \ldots, x_n)$. A value of the parameter θ that maximizes $L(\theta)$ is called a maximum likelihood estimator (MLE), and is denoted by θ_{ML}. It is often easier to maximize the log-likelihood function, $\log L(\theta)$, and since the (natural) logarithmic function is monotonically increasing in θ, the same value of θ_{ML} maximizes both $L(\theta)$ and $\log L(\theta)$. Under quite general conditions, MLEs have a number of favourable properties. Consistency: Under mild conditions, MLEs converge to the true parameter value as the sample size increases. Asymptotic Normality: As the sample size increases, the distribution of the MLE approaches that of a (potentially) Multivariate Normal variables.

cal distributions of the simulated data series are Exponential, Gamma, Log-Normal and Mixed Log-Normal distributions. The detailed information about these distributions can be found in Appendix A.

Columns three and four report the K-S Test Statistics and their P-Values. Although we display Q-Q plots to compare empirical distribution to theoretical distributions whether these two population distributions are exactly the same, we also conduct a statistical test to prove it. Checking by eye, the quantiles for the first distribution versus the quantiles for the second distribution will fall on the 0 – 1 line of the Q-Q plots can be insufficient. It can be both difficult and subjective to decide how differences between distributions will yield various kinds of deviations from a straight line. Appendix B presents details about the probability plots or Q-Q plots.

K-S Test Statistics belong to the goodness of fit tests which indicate whether or not it is reasonable to assume that a random sample comes from a specific distribution. They are a form of hypothesis testing where the null and alternative hypotheses are:

- H_0: the data follow a specified distribution
- H_A: the data do not follow the specified distribution

The K-S test is used to decide if a sample comes from a population with a specific distribution. It can be applied both for discrete (count) data and continuous binned and both for continuous variables. It is based on a comparison between the empirical distribution function (ECDF) and the theoretical one that is the upper extreme among absolute value differences between ECDF and the theoretical CDF.

The hypothesis regarding the distributional form is rejected if the K-S Test Statistic, KSTS, is greater than the critical value obtained from a table, or, which is the same, if the P-value is lower than the significance level.

For example in Table 3.2 for 10 nodes and employing no MAC protocol, the K-S Test Statistic, $KSTS = 0.09$, P-value $= 0.13$ alternative hypothesis is two sided. These values are obtained as means of KSTS values and P-values of 87 runs starting from the Lower Confidence Level value of the estimated rate parameter of

Exponential distribution to the Upper Confidence Values. It means that we cannot reject null hypothesis that the data follow an Exponential distribution because the P-value is enough higher than significance levels usually referred in statistical literature. The Figure 3.2 represents the histogram of inter-arrival times considered in this case. Figure 3.3 represents the QQ-plot for the exponential distribution, Figure 3.4 represents the empirical and theoretical exponential PDF, while Figure 3.5 represents the empirical and theoretical CDF.

Similarly, for Table 3.3 for 10 nodes, with T-MAC employed and sending 1 packet every 5 minutes, the $KSTS = 0.11$, P-value $= 0.15$. These values are obtained as means of KSTS values and P-values of 100 runs starting from the Lower Confidence Level value of the estimated rate parameter of Exponential distribution to the Upper Confidence Values. The corresponding theoretical distribution for the empirical one is Mixed Log-Normal. The Figure 3.6 represents the histogram of inter-arrival times considered in this case. The Figure 3.7 represents the histogram of inter-arrival times for the first part of the histogram considered in this case, while Figure 3.8 represents the histogram of inter-arrival times for the second part of the histogram considered in this case. Figure 3.9 represents the QQ-plot for the Mixed Log-Normal distribution, Figure 3.10 represents the empirical and theoretical Mixed Log-Normal PDF, while Figure 3.11 represents the empirical and theoretical Mixed Log-Normal CDF.

When CSMA protocol is employed for 20 nodes, sending 1 packet every second, Table 3.8 is presented giving the distribution details. The $KSTS = 0.006$, P-value $= 4.60 * e^{-05}$ are presented for an average of 153 runs. The corresponding theoretical distribution for the empirical one is non-Exponential distribution. Figure 3.35 represents histogram of inter arrival times, Figure 3.36 represents the QQ-plot for exponential distribution. Figures 3.37 and 3.38 represents PDF and CDF of empirical and theoretical exponential distributions respectively.

As we see from the tables presented, the empirical exponential arrival distribution assumption of wireless sensor networks holds only for two cases: 10 nodes, one packet every 5 minutes without MAC and 10 nodes, one packet every 10 minutes with CSMA. There are both theoretically known such as Gamma, Log-normal and Mixed Log-Normal of arrival distributions and theoretically unknown such as non-Exponential and Mixed arrival distributions in WSNs. It seems by increasing the

number of nodes, the modes of empirical distributions are getting lower values. At the same time, the right tails of the distributions are getting higher values. In other words, the empirical distributions are squeezed and pushed to the right having tails from Exponential to Gamma and then to Log-Normal distributions. If there are discontinuities of the empirical distributions then mixed theoretical distributions look more proper such as Mixed Log-Normal distribution of 10 nodes, one packet every 5 minutes with TMAC and 10 nodes, one packet every 5 seconds with CSMA. However, finite mixture models are often over-parametrized, leading to identification issues such as in 20, 35 and 40 nodes with TMAC and 20 nodes, one packet every 5 seconds of CSMA where the distributions are mixed but theoretically unknown.

When CSMA/CA is employed as the MAC protocol, for low data rates and lower number of nodes (10 nodes sending 1 packet every 10 minutes), the corresponding theoretical distribution for the empirical one is exponential (Table 3.4). This is due to the exponential back-off. As the number of nodes increases and as the data rate increases to 1 packet every 1 second, characterization of the arrival distributions at the CH becomes arduous due to the delay caused while the nodes try to repeatedly transmit the RTS beacon packets. Maximum Likelihood parameters of empirical distributions are estimated, theoretical distributions based on the estimated parameters are then generated. K-S Test Statistics for each generated data series are conducted in order to verify if it is possible to have a corresponding theoretical distribution. Due to the effects of CSMS/CA, arrivals at the CH follow theoretically unknown mixed distributions (Tables 3.7, 3.8, 3.9).

For the cases where TMAC is employed as MAC protocol, though the energy consumption is low by reducing the amount of energy wasted on idle listening by placing an adaptive duty cycle, there is extra delay incurred as the node takes an extra amount of time to wake up. Hence, characterizing arrival distribution at the cluster head accurately becomes complex. When any node has a packet to send, it starts to repeatedly transmit RTS beacon packets based on CSMA manner, i.e. through carrier sense and random back-off manner, making sure the channel is idle before it can transmit a packet, therefore causing delays. Although this can save considerable amounts of energy, this extra delay slightly increases the probability of collision. From the results presented in Tables 3.3, 3.6, 3.11, 3.13 and

their corresponding figures, it is evident that the effects of TMAC on the arrival distributions are clear. Except for the case where the number of nodes is relatively smaller, where the corresponding distribution follows Mixed Log-Normal, in all the other cases the corresponding theoretical distribution for the empirical one follows an unknown mixed distribution.

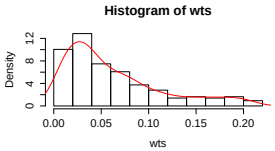

Figure 3.2: Histogram of Inter arrival times

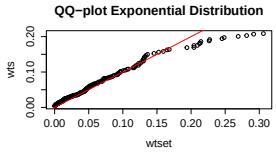

Figure 3.3: QQ-plot for Exponential Distribution

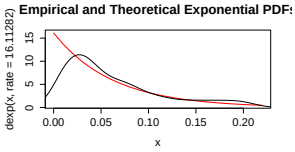

Figure 3.4: Empirical and Theoretical Exponential PDF

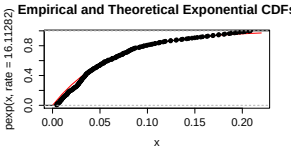

Figure 3.5: Empirical and Theoretical Exponential CDF

Table 3.2: Distribution of Inter-Arrival times, for 10 nodes with no MAC, sending 1 packet every 5 minutes; corresponding Figures 3.2, 3.3, 3.4, 3.5

Number of observation in the simulated series	ML Estimates of the parameters of empirical distribution	Kolmogorov-Smirnov Test statistics	P-values	Corresponding theoretical distribution for the empirical one
All:246 Used:214	Exponential rate = 16.11 LCL = 14.03 UCL = 18.03	Average of 87 runs (from LCL to UCL):0.09	Average of 87 runs (from LCL to UCL):0.13	Exponential

Table 3.3: Distribution of Inter-Arrival times, for 10 nodes with TMAC, sending 1 packet every 5 minutes; corresponding Figures 3.6, 3.7, 3.8, 3.9, 3.10, 3.11

Number of observation in the simulated series	ML Estimates of the parameters of empirical distribution	Kolmogorov-Smirnov Test statistics	P-values	Corresponding theoretical distribution for the empirical one
All:279 Used:249 First part: 224 Second part: 23	Mixed Log-Normal Meanlog1 = -5.14 Sdlog1 = 0.23 Meanlog2 = -0.52 Sdlog2 = 0.02 Mixing proportion: 0.09	Average of 100 runs :0.11	Average of 100 runs :0.15	Mixed Log-Normal

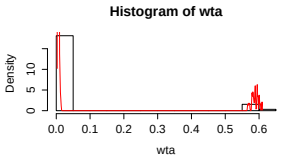

Figure 3.6: Histogram of Inter arrival times

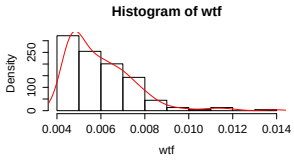

Figure 3.7: Histogram of Inter arrival times of first part

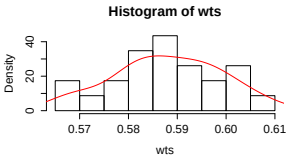

Figure 3.8: Histogram of Inter arrival times of second part

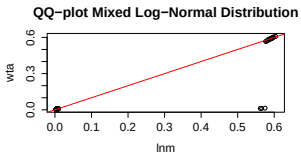

Figure 3.9: QQ-plots of Mixed Log-Normal Distribution

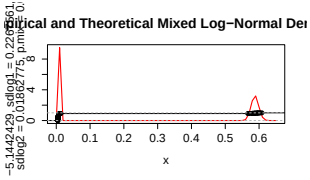

Figure 3.10: Empirical and Theoretical Mixed Log-Normal PDF

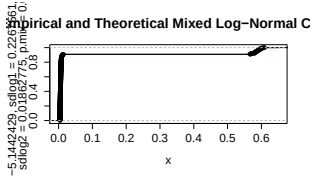

Figure 3.11: Empirical and Theoretical Mixed Log-Normal CDF

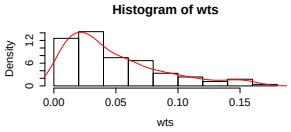

Figure 3.12: Histogram of Inter arrival times

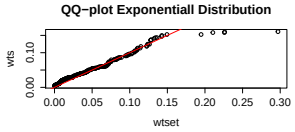

Figure 3.13: QQ-plot for Exponential Distribution

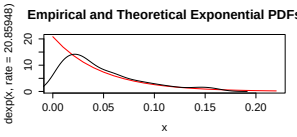

Figure 3.14: Empirical and Theoretical Exponential PDF

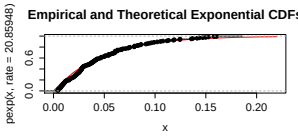

Figure 3.15: Empirical and Theoretical Exponential CDF

Table 3.4: Distribution of Inter-Arrival times, for 10 nodes with CSMA, sending 1 packet/10 minutes; corresponding Figures 3.12, 3.13, 3.14, 3.15

Number of observation in the simulated series	ML Estimates of the parameters of empirical distribution	Kolmogorov-Smirnov Test statistics	P-values	Corresponding theoretical distribution for the empirical one
All:255 Used:255	Exponential Rate = 20.86 LCL = 18.38 UCL = 23.50	Average of 141 runs :0.04	Average of 141 runs :0.45	Exponential

Table 3.5: Distribution of Inter-Arrival times, 20 nodes without MAC, sending 1 packet every 5 minutes; corresponding Figures 3.16, 3.17, 3.18, 3.19

Number of observation in the simulated series	ML Estimates of the parameters of empirical distribution	Kolmogorov-Smirnov Test statistics	P-values	Corresponding theoretical distribution for the empirical one
All:443 Used:411	Gamma shape = 1.49 scale = 0.03	Average of 100 runs :0.08	Average of 100 runs :0.24	Gamma

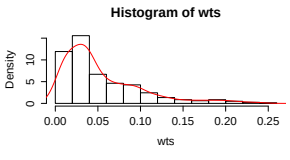

Figure 3.16: Histogram of Inter arrival times

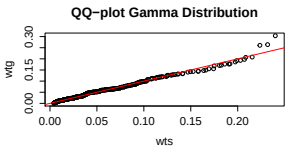

Figure 3.17: QQ-plot for Gamma Distribution

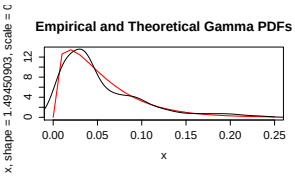

Figure 3.18: Empirical and Theoretical Gamma PDF

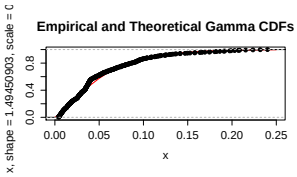

Figure 3.19: Empirical and Theoretical Gamma CDF

Table 3.6: Distribution of Inter-Arrival times for 20 nodes, with TMAC, sending 1 packet every 5 minutes; corresponding Figures 3.20, 3.21, 3.22, 3.23, 3.24, 3.25, 3.26

Number of observation in the simulated series	ML Estimates of the parameters of empirical distribution	Kolmogorov-Smirnov Test statistics	P-values	Corresponding theoretical distribution for the empirical one
All:574 Used:542 First part: 508 Second part: 34	Mixed Log-Normal Meanlog1 = -5.18 Sdlog1 = 0.25 Meanlog2 = -0.58 Sdlog2 = 0.04 Mixing proportion: 0.06	Average of 100 runs :0.16	Average of 100 runs : $7.66 * e^{-06}$	An unknown Mixed distribution

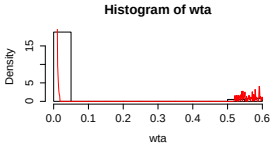

Figure 3.20: Histogram of Inter arrival times

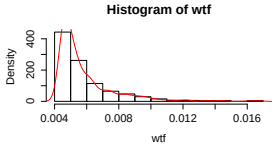

Figure 3.21: Histogram of Inter arrival times first part

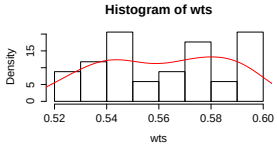

Figure 3.22: Histogram of Inter arrival times second part

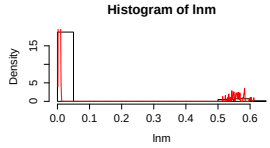

Figure 3.23: Histogram of log-normal distribution

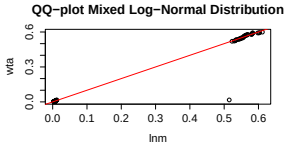

Figure 3.24: QQ-plot of Mixed Log-Normal Distribution

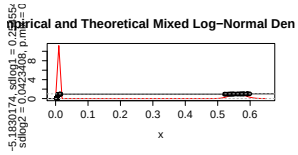

Figure 3.25: Empirical and Theoretical Mixed Log-Normal Densities

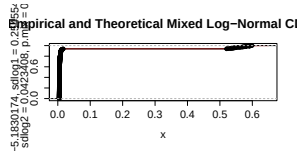

Figure 3.26: Empirical and Theoretical Mixed Log-Normal CDF

Table 3.7: Distribution of Inter-Arrival times, for 10 nodes with CSMA, sending 1 packet every 5 seconds; corresponding Figures 3.27, 3.28, 3.29, 3.30, 3.31, 3.32, 3.33, 3.34

Number of observation in the simulated series	ML Estimates of the parameters of empirical distribution	Kolmogorov-Smirnov Test statistics	P-values	Corresponding theoretical distribution for the empirical one
All:596 Used:596 First part: 578 Second part: 18	Mixed Log-Normal Meanlog1 = -3.78 Sdlog1 = 1.07 Meanlog2 = 1.59 Sdlog2 = 0.008 Mixing proportion: 0.03	Average of 100 runs :0.09	Average of 100 runs : 0.035	Mixed Log-Normal at p-values 3.5% or less. It is not Mixed Log-Normal at traditional 5 or 10% significance levels

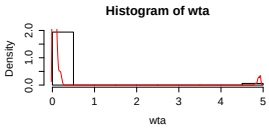

Figure 3.27: Histogram of Inter arrival times

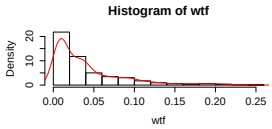

Figure 3.28: Histogram of Inter arrival times first part

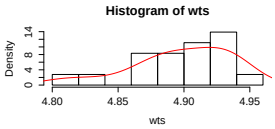

Figure 3.29: Histogram of Inter arrival times second part

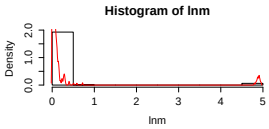

Figure 3.30: Histogram of log-normal distribution

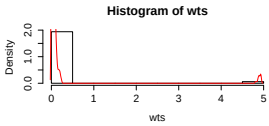

Figure 3.31: QQ-plot of Mixed Log-Normal Distribution

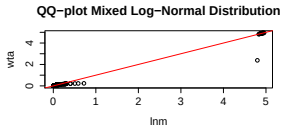

Figure 3.32: QQ-plot for Mixed Log-Normal Distribution

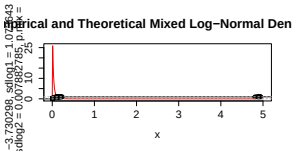

Figure 3.33: Empirical and Theoretical Mixed Log-Normal Densities

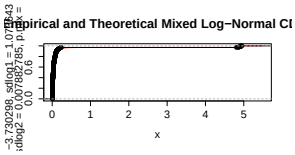

Figure 3.34: Empirical and Theoretical Mixed Log-Normal CDF

Table 3.8: Distribution of Inter-Arrival times, for 20 nodes with CSMA, sending 1 packet every 1 second; corresponding Figures 3.35, 3.36, 3.37, 3.38

Number of observation in the simulated series	ML Estimates of the parameters of empirical distribution	Kolmogorov-Smirnov Test statistics	P-values	Corresponding theoretical distribution for the empirical one
All:1889 Used:1889	Exponential rate = 16.91 LCL = 16.91 UCL = 17.69	Average of 153 runs (from LCL to UCL):0.06	Average of 153 runs (from LCL to UCL):$4.60 * e^{-05}$	An unknown non-Exponential distribution

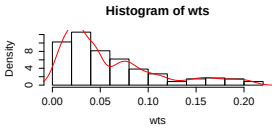

Figure 3.35: Histogram of Inter arrival times

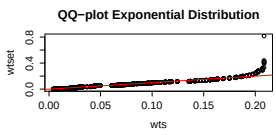

Figure 3.36: QQ-plot for Exponential Distribution

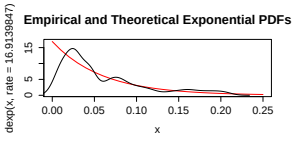

Figure 3.37: Empirical and Theoretical Exponential PDF

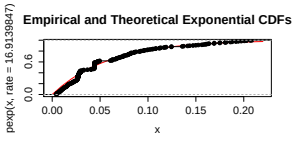

Figure 3.38: Empirical and Theoretical Exponential CDF

Table 3.9: Distribution of Inter-Arrival times for 20 nodes, with CSMA, sending 1 packet every 5 seconds; corresponding Figures 3.39, 3.40, 3.41, 3.42, 3.43, 3.44, 3.45

Number of observation in the simulated series	ML Estimates of the parameters of empirical distribution	Kolmogorov-Smirnov Test statistics	P-values	Corresponding theoretical distribution for the empirical one
All:2010 Used:2010 First part: 1990 Second part: 20	Mixed Log-Normal Meanlog1 = -2.94 Sdlog1 = 1.05 Meanlog2 = -0.10 Sdlog2 = $0.2 * e^{-04}$ Mixing proportion: 0.001	Average of 100 runs :0.06	Average of 100 runs : $2.62 * e^{-03}$	An unknown Mixed distribution

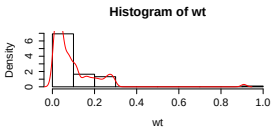

Figure 3.39: Histogram of Inter arrival times

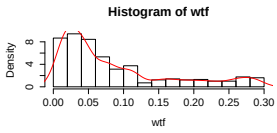

Figure 3.40: Histogram of Inter arrival times first part

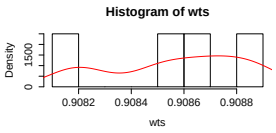

Figure 3.41: Histogram of Inter arrival times second part

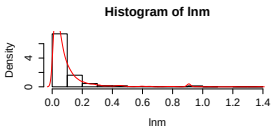

Figure 3.42: Histogram of log-normal distribution

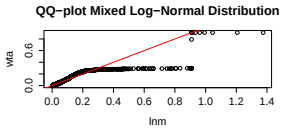

Figure 3.43: QQ-plot of Mixed Log-Normal Distribution

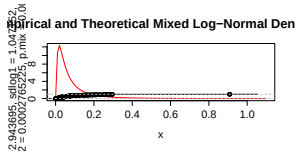

Figure 3.44: Empirical and Theoretical Mixed Log-Normal Densities

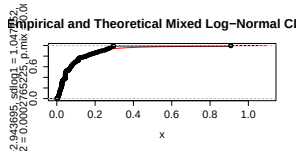

Figure 3.45: Empirical and Theoretical Mixed Log-Normal CDF

Table 3.10: Distribution of Inter-Arrival times, for 35 nodes with no MAC, sending 1 packet every 5 minutes; corresponding Figures 3.46, 3.47, 3.48, 3.49

Number of observation in the simulated series	ML Estimates of the parameters of empirical distribution	Kolmogorov-Smirnov Test statistics	P-values	Corresponding theoretical distribution for the empirical one
All:644 Used:612	Log-Normal Meanlog = -3.59 Sdlog=0.93 LCL = -3.66 UCL = -3.52	Average of 148 runs (from LCL to UCL):0.08	Average of 148 runs (from LCL to UCL):0.13	Log-Normal

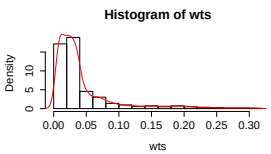

Figure 3.46: Histogram of Inter arrival times

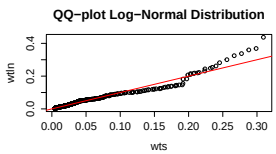

Figure 3.47: QQ-plot for Log-Normal Distribution

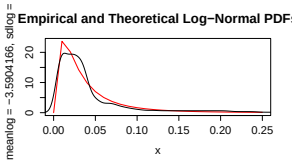

Figure 3.48: Empirical and Theoretical Log-Normal PDF

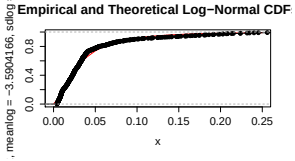

Figure 3.49: Empirical and Theoretical Log-Normal CDF

Table 3.11: Distribution of Inter-Arrival times for 35 nodes, with TMAC, sending 1 packet every 5 minutes; corresponding Figures 3.50, 3.51, 3.52, 3.53, 3.54, 3.55, 3.56

Number of observation in the simulated series	ML Estimates of the parameters of empirical distribution	Kolmogorov-Smirnov Test statistics	P-values	Corresponding theoretical distribution for the empirical one
All:990 Used:958 First part: 914 Second part: 44	Mixed Log-Normal Meanlog1 = -5.23 Sdlog1 = 0.23 Meanlog2 = -0.62 Sdlog2 = 0.09 Mixing proportion: 0.05	Average of 100 runs :0.19	Average of 100 runs : $3.18 * e^{-14}$	An unknown Mixed distribution

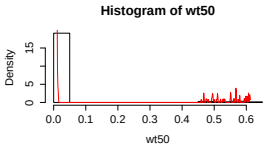

Figure 3.50: Histogram of Inter arrival times

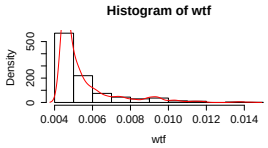

Figure 3.51: Histogram of Inter arrival times first part

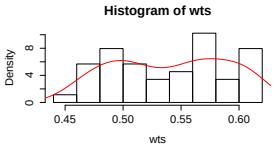

Figure 3.52: Histogram of Inter arrival times second part

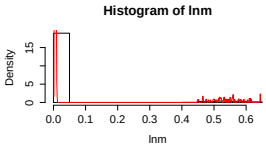

Figure 3.53: Histogram of log-normal distribution

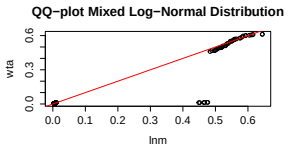

Figure 3.54: QQ-plot of Mixed Log-Normal Distribution

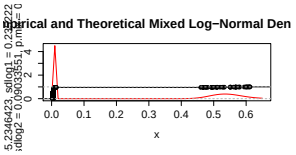

Figure 3.55: Empirical and Theoretical Mixed Log-Normal Densities

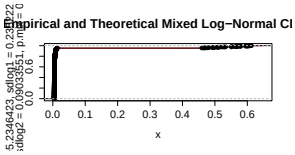

Figure 3.56: Empirical and Theoretical Mixed Log-Normal CDF

Table 3.12: Distribution of Inter-Arrival times, for 40 nodes with no MAC, sending 1 packet every 5 minutes; corresponding Figures 3.57, 3.58, 3.59, 3.60

Number of observation in the simulated series	ML Estimates of the parameters of empirical distribution	Kolmogorov-Smirnov Test statistics	P-values	Corresponding theoretical distribution for the empirical one
All:751 Used:720	Log-Normal Meanlog = -3.77 Sdlog=0.90 LCL = -3.84 UCL = -3.71	Average of 100 runs (from LCL to UCL):0.08	Average of 100 runs (from LCL to UCL):0.07	Log-Normal at p-values 6% or less. it is not Log-Normal at traditional 10% significance level

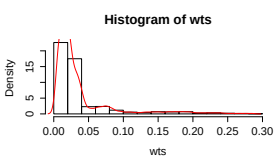

Figure 3.57: Histogram of Inter arrival times

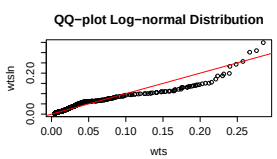

Figure 3.58: QQ-plot for Log-Normal Distribution

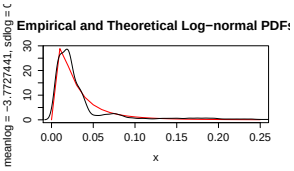

Figure 3.59: Empirical and Theoretical Log-Normal PDF

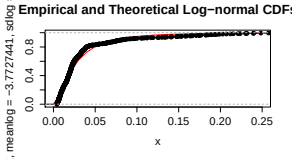

Figure 3.60: Empirical and Theoretical Log-Normal CDF

Table 3.13: Distribution of Inter-Arrival times for 40 nodes, with TMAC, sending 1 packet every 5 minutes; corresponding Figures 3.61, 3.62, 3.63, 3.64, 3.65, 3.66

Number of observation in the simulated series	ML Estimates of the parameters of empirical distribution	Kolmogorov-Smirnov Test statistics	P-values	Corresponding theoretical distribution for the empirical one
All:1138 Used:1106 First part:1064 Second part: 38	Mixed Log-Normal Meanlog1 = -5.26 Sdlog1 = 0.22 Meanlog2 = -0.62 Sdlog2 = 0.10 Mixing proportion: 0.04	Average of 100 runs :0.22	Average of 100 runs : 0.0	An unknown Mixed distribution

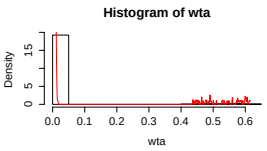

Figure 3.61: Histogram of Inter arrival times

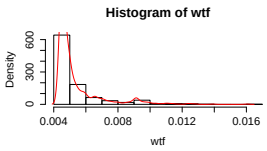

Figure 3.62: Histogram of Inter arrival times first part

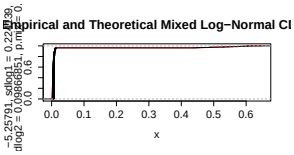

Figure 3.63: Empirical and Theoretical Mixed Log-Normal CDF

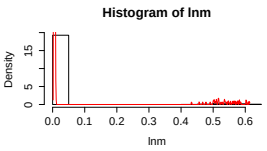

Figure 3.64: Histogram of log-normal distribution

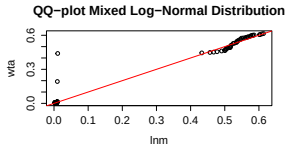

Figure 3.65: QQ-plot of Mixed Log-Normal Distribution

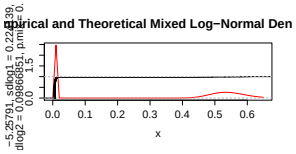

Figure 3.66: Empirical and Theoretical Mixed Log-Normal Densities

3.7 Summary

Wireless Sensor Networks have seen a tremendous growth in various application areas despite prominent performance and availability challenges. Models for sensor networks have developed quickly and are now much more sophisticated than they were some years ago. However, we believe that the quest for new models is still of prime importance. In fact, it involves several research areas, for example modulation techniques, channel modelling, queuing theory, traffic modelling, network

protocol design and optimization techniques. Although researchers continue to address these challenges, the type of data arrival distributions at the cluster head and intermediary routing nodes is still an interesting area of investigation. The general practice in published works is to compare an empirical exponential arrival distribution of wireless sensor networks with a theoretical exponential distribution in a Q-Q plot diagram. In this chapter, results presented show that such comparisons based on simple eye checks are not sufficient since, in many cases, incorrect conclusions may be drawn from such plots. After estimating the Maximum Likelihood parameters of empirical distributions, we generate theoretical distributions based on the estimated parameters

To the best of our knowledge, this is the first work that provides statistical proof for finding theoretical distributions of arrivals at the CH and relay nodes in WSNs. Each node is modelled according to a queuing model and is characterised by its sending rate, inter-arrival distribution and the service process. The empirical distributions of inter-arrival times of the packets considering such physical events that do not occur frequently are generally assumed by Poisson processes, and the inter-arrival times by exponential distributions. The general practice in published works is thus to compare empirical exponential arrival distributions of wireless sensor networks with theoretical exponential distributions in Q-Q plot diagrams. In this chapter, we show that such comparisons based on simple eye checks are not sufficient since in many cases incorrect conclusions may be drawn from such plots. After estimating Maximum Likelihood parameters of empirical distributions, we generate theoretical distributions based on the estimated parameters. By conducting Kolmogorov-Smirnov Test Statistics for each generated data series, we find out, if it is possible, a corresponding theoretical distribution. Empirical exponential arrival distribution assumption of wireless sensor networks holds only for a few cases. There are both theoretically known such as Gamma, Log-normal and Mixed Log-Normal of arrival distributions and theoretically unknown such as non-Exponential and Mixed arrival distributions in wireless sensor networks. The effects caused by MAC properties are also analysed by experimenting with well known MAC protocols and the summary of the inter arrival time distributions after extensive tests are presented for various application categories in 3.14. Therefore, these results confirm that the assumption of exponential inter-arrival distributions

does not hold in all the cases. Exponential arrival distribution assumption of wireless sensor networks holds only when a fewer nodes (10-15), sending packet every 5-10 minutes with no MAC properties, as-well as when CSMA properties are considered.

Table 3.14: Summary for Inter arrival time distributions for various application categories

S.No	Application Type	Packet Rate (1packet/time)	Inter arrival time distribution		
			No MAC	TMAC	CSMA
1	Environment monitoring, Smart agriculture	5 - 10 min	Exponential for 10-15 nodes	Mixed-Log normal for 10-15 nodes	Exponential for 10-15 nodes
2	Traffic monitoring, Vehicle tracking	5 - 10 sec	Gamma for 20-30 nodes	Unknown Mixed distribution	Unknown mixed distribution
3	Military applications, Body Area Networks	1 sec or higher	Constant	Unknown mixed distribution	Unknown non-Exponential

Chapter 4

Architectural Framework for WSN

4.1 Introduction

In this chapter, an architecture-driven modelling platform for the development and the analysis of WSNs is presented. The modelling viewpoints and conceptual elements have been carefully designed in collaboration with colleagues from various domains, such as software engineering, wireless sensor networks, and telecommunications as their design and implementation requires corporation of different backgrounds. The expertise of the distinguished researchers from University of L'Aquila in the areas of software engineering highly motivated the development of various frameworks for the platform [1]. Together with the specifications of software architecture, fine-grain simulations are included that combine different information such as low-level details, hardware specifications and the physical environment. A detailed analysis on existing simulation tools is performed in order to incorporate the most realistic one in this study. The modelling views are analysed, combined and translated into low level simulation scripts. The programming framework functioning has been tested and expressivity of the framework has been validated by realizing a plug-in devoted to energy-related simulation of WSNs, by taking into account the physical environment and other factors. Popular scenarios including health care and home automation have been used as case studies in order to demonstrate the new architecture and also to show the effectiveness of the

[1] http://www.ivanomalavolta.com/

architectural approach and the plug-in developed.

A WSN consists of spatially distributed autonomous sensor nodes that cooperate in order to accomplish a specific task. Sensor nodes are cheap, small, and battery-powered devices with limited processing capabilities and memory. WSNs are mostly developed directly on the top of the operating system. They are tied to the hardware configuration of the sensor nodes and their design and implementation can require cooperation with a myriad of system stakeholders with different backgrounds. Despite the increasing usage of WSNs in modern applications, their development is still plagued by the following issues:

Development is still performed directly on the top of the operating systems and relies on individuals hard-earned programming skills across all levels of the communication stack (e.g., application, routing, data link levels, etc.) [Mottola and Picco, 2011];

WSN engineers must address challenging extra-functional requirements such as performance, security, energy consumption, with poor support for early testing, debugging, and simulation of WSNs in an integrated fashion [Imran et al., 2010];

In-order to achieve the demanded high level of efficiency, the software of a WSN application is tied to specific hardware platforms, thus hampering the reuse of source code and software components across different projects or organizations [Mottola and Picco, 2011];

Due to the intrinsic multidisciplinary nature of the WSN problem space, WSN engineers must continuously collaborate with a high number of system stakeholders (e.g., WSN users, application domain experts, hardware designers, and software developers) with different background and training [Romer and Mattern, 2004b].

This ameliorates the implementation and enables different analysis in terms of various factors, such as energy efficiency and performance evaluation to be performed at the early stages of WSN design and development. Approaches abstracting implementation details from the underlying hardware and physical infrastructure are strongly advised [Blumenthal et al., 2003; Mottola and Picco, 2011].

In the WSN community, there is a growing awareness of the need for methodologies, techniques, and abstractions that simplify development tasks and increase the

confidence in the correctness and performance of the resulting software. Software engineering (SE) support is therefore sought, not only to ease the development task but also to make it more reliable, dependable, and repeatable. For instance, authors in [Picco, 2010] point out the need of high-level abstractions that simplify the configuration of the WSN at large, possibly allowing one to define its software architecture based on existing components. As a possible solution to the above mentioned issues, the WSN community is becoming aware of the need of using software engineering approaches in order to support the design, analysis, simulation and implementation of WSNs [Picco, 2010; Willig, 2006a].

This research also contributes a novel modelling and analysis platform to support an architecture-driven development of WSNs. The platform is called A4WSN[1] and is based on a multi-view architectural approach [ISO/IEC/IEEE, 2011] based on three modelling languages to describe a WSN from different viewpoints, which can be summarised as follows:

- Software components and their interactions,
- The low-level and hardware specification of sensor nodes, and
- The physical environment where sensor nodes are deployed, separately.

Model-driven engineering (MDE) techniques and tools are used to realise the modelling framework through metamodelling, model weaving and model transformation. The modelling framework is supported by a programming framework that enables the implementation of analysis and code generation plugins by third party developers; they can be employed to assess and analyse the architectural design decisions and used to generate executable code, respectively.

The whole A4WSN platform is realised by exploiting advanced Model-driven Engineering (MDE) techniques, such as metamodelling, model weaving and model transformation. MDE allows the user to define the modelling languages of A4WSN in a seamless and well-disciplined manner, and to realise the A4WSN programming framework so that it supports extensibility and customisation by design. This work provides evidence on the applicability of the proposed modelling approach and on

[1] A4WSN stands for **A**rchitecting platform for (**4**) **W**ireless **S**ensor **N**etworks

the extensibility of its programming framework by developing a A4WSN plugin called PlaceLife. Placelife analyses A4WSN models of a WSN and automatically assesses the life time of the network. The simulations are more realistic compared to other simulation tools which use a simplified model of the environment (e.g., a free space model). Clustering techniques, security protocols can also be analysed with the help of Placelife and the performance of the WSN can be accurately estimated. PlaceLife uses the A4WSN physical environment model that includes physical objects and material, thus providing an accurate WSN lifetime estimation. The *PlaceLife* plugin has been applied onto a realistic case study in the health-care application domain.

The main contributions of this chapter can be summarised as follows:

- A multi-view modelling platform for engineering WSNs is presented (including a specification of the software architecture, low-level and hardware specification, and physical environment);
- A programming framework is provided which enables the development of plugins realising new code generation and analysis engines;
- The A4WSN prototype tool[1] realising the approach presented has been implemented; it is based on the Eclipse platform and can be integrated with other MDE technologies already available in the Eclipse community;
- The *PlaceLife* plugin for the prediction of the life time of a WSN by taking into account also the physical environment (together with other factors) is presented.

4.2 A4WSN Overview

In this section, an overview of the A4WSN platform is presented. As shown in Figure 4.1, the A4WSN platform is composed of two main parts: a modelling environment for describing the architecture of a WSN and a programming framework.

[1] A4WSN prototype: http://a4wsn.di.univaq.it

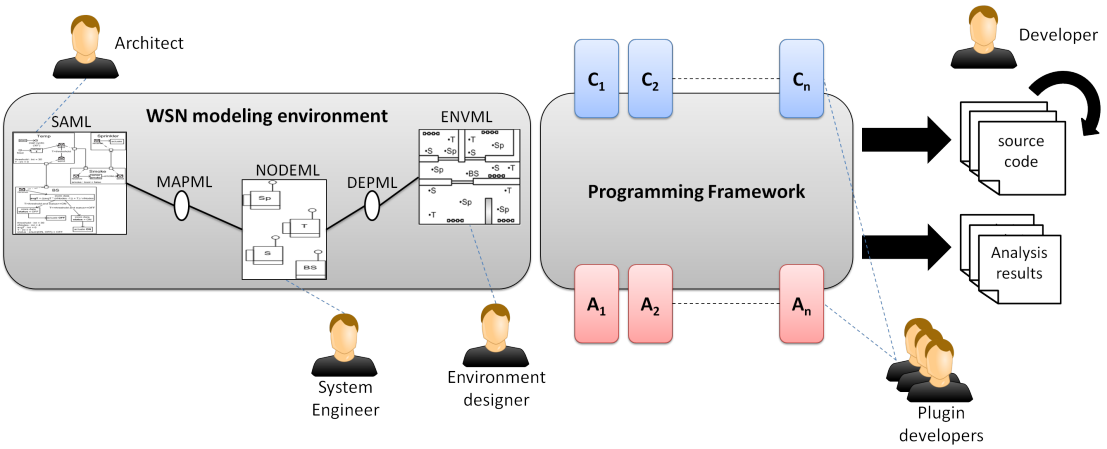

Figure 4.1: Overview of the A4WSN platform

The **modelling environment** exposes three modelling languages for describing specific architectural views of a wireless sensor network: the *Software Architecture Modelling Language for WSN* (*SAML*), the *Node Modeling Language (NODEML)*, and the *Environment Modelling Language (ENVML)*. In the following, the rationale driving factors that led to propose each modelling language is presented. The SAML language focuses on the *application layer* of the WSN. It is used to break down the application into a set of software entities (e.g., components), to show how they relate to each other, to better reason on their distribution throughout the network, and to reason on the business logic of the WSN. The NODEML language concerns all the low-level aspects underneath the application layer of the WSN. In this context, stakeholders reason about routing protocols, middleware, hardware configuration of the nodes, etc. The ENVML language is about the site in the real world where the WSN will be deployed. This viewpoint could be specially useful for developers and system engineers when they have to reason about the network topology, the presence of possible physical obstacles (e.g., walls, trees) within the network deployment area, and so on.

The programming framework provides a set of facilities for supporting the development and integration of either code generation or analysis engines. Each engine is realised as a plugin of this framework. The proposed programming framework knows at run-time which plugins are installed into the framework, and automatically provides to the user the available target implementation languages or

the available analysis techniques.

In order to provide a useful instrument for modelling WSNs, the system concerns are identified, which are held by engineers and developers while designing a WSN. This set of concerns is based on the literature, experiences in the field of WSN development and on some informal investigation performed on how expert engineers and developers have worked on previous WSN projects. The resulting set of system concerns in the domain of WSN can be summarised as:

- *Energy Consumption* concerns how much power is consumed by the application running on the nodes with respect to the used batteries or harvested energy sources. It is a crucial factor that constrains the networks life time for the WSN mission.
- *Dependability* is a generic attribute to describe "the trustworthiness of a computing system which allows reliance to be justifiably placed on the service it delivers". Dependability includes attributes such as reliability, availability, safety, security as special cases[1].
- *Coverage* represents how well an area is monitored or tracked by sensors [Huang and Tseng, 2003].
- *Networking and Communication* represent the various networking aspects which may affect the WSN. Example of the issues which are related to this specific stakeholder comprise: routing protocols, MAC protocols, usage of a specific communication middleware, connectivity of the network, etc.
- *Performance* deals with all performance-related aspects of the WSN, such as throughput, latency, response times, nodes utilisation, etc.
- *Data Collection* concerns data structures and how data is collected, managed and transformed by the various nodes within the WSN.

We designed the proposed modelling languages so that they can frame all the above listed concerns. The three proposed modelling languages are linked together via two auxiliary modelling languages in order to create a combined software,

[1] http://www.dependability.org/

nodes, and environmental view of a WSN. These languages are called Mapping Modelling Language (*MAPML*) and Deployment Modelling Language (*DEPML*), and they link together SAML to NODEML and NODEML to ENVML, respectively. More specifically, MAPML links are used for assigning software components to the corresponding hardware node configuration they will be executed on. Whereas, DEPML links are used for virtually deploying WSN nodes into specific areas within the physical environment. We decided to use those two auxiliary modelling languages for a clear *separation of concerns* and duties while architecting the WSN (e.g., a software architect can focus on the application layer in the SAML model only, while a system engineer may focus on the nodes configurations in the NODEML model) and making the used models *reusable* across projects and organizations (for example, the same nodes configuration defined in a NODEML model can be shared across different applications produced by a software company). The main concepts of each modelling language will be described in Section 4.2.1.

The **programming framework** provides a set of facilities for supporting the development and integration of either *code generation* or *analysis* engines. Each engine is realised as a plugin of this framework. Our proposed programming framework knows at run-time which plugins are installed into the framework, and automatically provides to the user the available target implementation languages or the available analysis techniques.

Code generation and analysis plugins are structurally similar. An analysis plugin manages the analysis of WSNs (e.g., coverage, connectivity, energy consumption analysis), instead of a code generation plugin which is tailored to the generation of implementation code conforming to a set of specific target languages. More specifically, in the main difference between code generation and analysis plugins resides in their returned output: the main output of a code generation engine can either be a set of source files, or binary packages, whereas the main output of an analysis engine can be a violated property, a counter-example, a set of numerical values, and so on. The detailed description of the programming framework is presented in Section 4.2.2.

The platform is generic since it is independent from the programming language, hardware and network topology. Starting from a set of models (each one reflecting a certain WSN viewpoint), the code generation and analysis components can be

plugged into the framework for generating executable code or analysing outcomes.

4.2.1 The Modelling Environment

As shown in the previous section, the modelling environment is composed of three main languages, which are SAML, NODEML and ENVML. Each language allows the user to frame the problem of describing the architecture of a WSN from a specific viewpoint [ISO/IEC/IEEE, 2011].

More specifically, the **SAML** modelling language focuses on the application layer of the WSN. It is used to break the application down into a set of software entities (e.g., components), to show how they relate to each other, to better reason on their distribution throughout the network, and to reason on the business logic of the WSN. The **NODEML** modelling language concerns the low-level details underneath the application layer of the WSN. In this context, stakeholders reason about routing protocols, middleware, hardware configuration of the nodes, etc. The **ENVML** modelling language is about the physical environment where the WSN will be deployed. This viewpoint could be specially useful when they have to reason about the network topology, the existence of possible physical obstacles (e.g., walls, trees) within the network deployment area, and so on. The **MAPML** modelling language weaves together an SAML model and a NODEML model. It allows designers to define a set of mapping links, each of them weaving together components in the SAML model and node definitions in the NODEML model. The **DEPML** modelling language weaves a NODEML model to an ENVML model. It allows designers to consider each node type defined in the NODEML model and to *instantiate* it in a specific area within the physical environment defined in an ENVML model. Each node type can be instantiated any number of times within a specific area with a certain distribution strategy.

4.2.2 The Programming Framework

The A4WSN platform is composed of two main parts that are a modelling environment to allow architects to model WSN applications and a programming framework devoted to code generation and analysis of WSN application models. The motivation for performing code generation and analysis of WSN application

models are well understood both in academia and in practice [Lorincz et al., 2004; Picco, 2010]. Basically, code generation helps in reducing the cost of developing a WSN application since the developers can automatically obtain an executable application from the model by applying some specific transformations. Also, performing analysis is fundamental while developing a WSN application due to the intrinsic complexity of the WSN domain. For example, if typical aspects are considered in WSN development such as nodes connectivity, real-time communication, energy consumption, performance, security, etc., it is extremely difficult and demands for a lot of effort to ensure that a developed WSN is correct with respect to those aspects. Moreover, analysis engines can also be used to reason on the WSN configuration in order to find reasonable trade-offs in terms of network topology, employed protocols, etc. for a specific task.

The programming framework provides a generic workbench and a set of extension points for supporting the development and integration of third-party code generation and analysis engines. More specifically, through its components, it enables the storage of WSN models, supports the merging of linked models, validates A4WSN models, provides error/warning/information messages to the user, defines a UI manager to make plugins interacting, provides facilities for managing code generation and analysis engines. Third-party engines are realised as plugins extending the A4WSN generic workbench. It knows at run-time which plugins are available and automatically provides to the user the available target implementation languages and analysis techniques.

Figure 4.2 shows an overview of the A4WSN programming framework. All the boxes within the programming framework represent the various components of the generic programming workbench, whereas the $C_1..C_n$ and $A_1..A_n$ boxes represent third-party code generation and analysis plugins, respectively. Third-party plugins extend the *Code Generation Manager* and *Analysis Manager* components which provide the needed extension points and they communicate with all the other components of the programming framework (for the sake of clarity, those connectors are not shown in the figure)

Code generation and analysis plugins are structurally similar. Basically, an analysis plugin manages analyses for WSN (e.g., coverage, connectivity, energy consumption analysis), instead of a code generation plugin which is tailored to

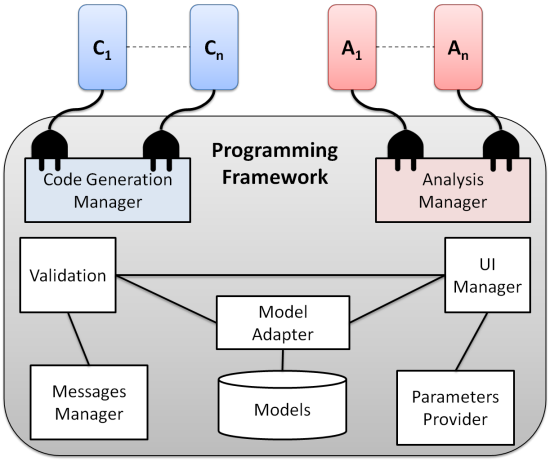

Figure 4.2: The A4WSN programming framework

the generation of implementation code conforming to a set of specific target languages. More specifically, in A4WSN the main difference between code generation and analysis plugins resides in their returned output: the main output of a code generation engine can either be a set of source files, or binary packages, whereas the main output of an analysis engine can be a violated property, a counter-example, a set of numerical values, and so on.

The A4WSN platform is generic as it is independent from the programming language, hardware and network topology. Starting from a set of models (each one reflecting a certain WSN viewpoint), the code generation and analysis components can be plugged into the framework for generating executable code or analysis outcomes.

4.2.3 Prototype Implementation

The current prototype of the proposed approach is made available to the community as an open-source product in order to allow other researchers to use the modelling languages introduced in Section 4.2.1 as well as the programming framework described in Section 4.2.2. The current prototype of A4WSN can be downloaded from the A4WSN website (http://a4wsn.di.univaq.it).

The proposed approach has been implemented by extending the **Eclipse** platform[1]. Eclipse is an open-source development platform comprised of extensible

[1]Eclipse project Web site: www.eclipse.org.

frameworks and tools for building, deploying and managing software across the life cycle. Eclipse has been chosen as starting point for the modelling environment for three main reasons. Firstly, many extensions already exist covering some aspects of this approach (e.g., graphical syntax definition for newly created languages, models persistence support, etc.). Secondly, its plugin architecture allows the user to provide a set of extension points that other developers can use to extend this modelling framework,. Thirdly, the Eclipse community is widely spread throughout the world, raising the possibility of adoption of this modelling environment.

For what concerns the modelling languages, model-driven engineering techniques are used to define their concepts, and their modelling environment. More specifically, the static semantics of the languages have been specified by means of their underlying metamodels. Those metamodels are defined by using the **Eclipse Modelling Framework** (EMF)[2], that is a Java framework and code generation facility for building tools and other applications based on a metamodel. The concrete syntax of the modelling languages has been defined by using the **Graphical Modelling Framework** (GMF)[3], a model-driven approach to generate graphical editors in Eclipse.

The intermediate modelling languages (i.e., MAPML and DEPML) are technically called weaving models. Weaving models are special kinds of models for defining relations among other models and to establish semantic links among model elements. Weaving models have been successfully used in many fields, such as software architecture [Malavolta et al., 2010] and software product lines [Czarnecki and Antkiewicz, 2005]. The **Atlas Model Weaver (AMW)** [Didonet Del Fabro M., Bézivin J., Jouault F. and Breton E. and Gueltas G., 2005] is used for managing those weaving models.

For what concerns the programming framework, this approach implemented it as a set of **Eclipse plugins**, each one implementing a single component of the programming framework, as it is depicted in Figure 4.2. Those plugins are implemented in Java and their dependencies are realized by means of the plugins management system provided by Eclipse. Each plugin declares the others it de-

[2]EMF project Web site: http://www.eclipse.org/modeling/emf/.
[3]GMF project Web site: http://www.eclipse.org/modeling/gmf/.

pends on and configuration parameters via a specific XML configuration file. The communication among plugins is handled by standard Java calls. Also, the code generation framework and the analysis framework provide two extension points dedicated to code generation and analysis plugins, respectively. The signatures of those extension points are defined in the same XML configuration files used for defining the dependencies between plugins, whereas their implementation is defined as Java classes referenced by the XML configuration files. For the sake of brevity, the details on how the programming framework works are not presented, and on how its plugins interact. A detailed description of those aspects can be found in the Eclipse plugin developer guide[1].

4.3 PlaceLife: an A4WSN plug-in

In order to validate the expressivity of the A4WSN modelling languages and to exercise the provided extension points, an analysis plug-in called PlaceLife has been developed. PlaceLife takes advantage of the three modelling views (namely, SAML, NODEML and ENVML) in order to provide an estimate of the WSN lifetime. All modelling views are analysed, combined and translated into low level simulation scripts that can be executed to estimate the WSN lifetime. This translation has been useful to verify that the proposed models have an appropriate level of details for simulation purposes. In order to produce a realistic simulation the following is desirable:

- **abstraction:** the models abstract all the details needed to generate scripts that can run in various well-accepted simulators such as Opnet and OMNET++;
- **fine-grain simulation:** the details should allow fine-grain simulations that combine different information such as physical environment, hardware and various layers of OSI.

The abstraction is verified by considering all OSI layers and for each layer the information required by well-established simulation tools. Fine-grain simulations

[1]Eclipse Platform Plug-in Developer Guide: http://help.eclipse.org/helios/index.jsp

are easily obtained thanks to the reuse and the weaving of multiple models into a single one. Models such as NODEML or ENVML that contain low level information (e.g., hardware and path loss) can be created once and reused multiple times with different application models. While expert users can write these complex low level models, average users can reuse them as many times as they like and concentrate on the application logic. This has been validated in Section 4.4.2 by means of a health care application. Technical details such as the effects of the path loss and the hardware which require theories of telecommunication are specified in pre built PlaceLife models. Technical details are complemented with the application model (i.e, SAML) and the physical environment (i.e., ENVML). These models are transformed into various complex simulation scripts. In Section 4.4, the PlaceLife implementation and the simulation tool used as a target language for simulation script generation (that is, Castalia) are explained. In section 4.4.2.1, the simulation results obtained are compared with a basic Castalia simulation written by an average user with a PlaceLife simulation based on pre-built hardware and path loss models. The former has the default ideal free space model for the path loss while the latter considers pre-built PlaceLife models that considers the real environment that is made of physical objects. Results show that by not considering the real environment cause overestimation of the lifetime which is particularly undesirable.

4.3.1 Application layer

Information at application layer should include the structure and behaviour of the WSN. For instance this includes type of components, number of instances and their interaction. This information is useful to derive relevant data such as the sensing rate, messages sent over the network and the type of communication (broadcast, multicast and point-to point). This data clearly affects the energy consumption. Beside the structure and the behaviour of the WSN, useful application layer information can be the used aggregation protocol[1] [Intanagonwiwat et al., 2002; Rajagopalan and Varshney, 2006] (if any), the type of operating system, the type of middleware (if any) and so on.

[1] Aggregation and fusion aims at removing redundant data and transmitting concise information.

The modelling languages of A4WSN provide ways to define all the aforementioned application layer information. The SAML view contains structure and behaviour of the WSN application. For instance this includes the type of components, their interaction, the sensing and transmission activities of a node and the type of transmission. This information is complemented by the NODEML view that specifies, among the other information, the type of operating system and the middleware used. In PlaceLife, the ENVML and DEPML models are used in order to have data about the number of nodes within the WSN and their deployment position in the environment. Application layer information is translated into low level scripts. More precisely, structure and behaviour are translated into simulation scripts. These scripts are combined with components from the simulation library (such as sensing components and middleware) in order to obtain the entire application layer configuration of the simulation. While PlaceLife provides the implementation of some libraries, unimplemented ones such as unknown sensors or unsupported middleware need to be specified by the user.

4.3.2 Network and data link layers

Information at the networking layer should specify the routing protocol employed. While routing can be performed in a multi-hop fashion, clustering approaches are very effective in order to improve the energy efficiency of the WSNs. This is why the NODEML can specify either multi-hop routing protocols (e.g., AODV) or some clustering approach (e.g., LEACH, HEED, In chapter 6, UHEED protocol is considered). A static routing can also be defined by explicitly specifying the connection among nodes. Routing that are not supported by A4WSN must be implemented by using some simulation script language.

Many challenges that arise because of the interactions between different layers can be addressed by altering the layered structure of the protocol architecture. The relation between the MAC and routing layers also impacts communication success. Since the wireless channel is essentially a broadcast medium, only a single transmission is allowed in a transmission area by the MAC protocol. As a result, simultaneous transfers are not possible. Moreover, the MAC layer introduces a non-deterministic delay for channel access because of the activities of other nodes.

As the routes constructed by the network layer constitute the source of contention, the MAC layer has to be aware of the multi-hop nature of the communication. If a neighbour of a node is transmitting a packet, the MAC protocol delays the transmission for a random amount of time to prevent collisions with the ongoing transmission as well as other neighbours that are trying to access the channel. This may significantly affect the performance of routing protocols, especially when distance or hop length measures are closely tied to the time in which a packet is received from a particular node.

Information at the data link layer should include the medium access control method (MAC). Access methods can be summarised into two main categories: contention based method (e.g., CSMA/CA) and channel partitioning (e.g., TDMA). The NODEML includes a wide range of possibilities for the MAC protocol selection. This includes CSMA, T-MAC and S-MAC [van Dam and Langendoen, 2003; Ye et al., 2004b].

4.3.3 Physical layer and hardware

Physical layer information should support the definition of an energy consumption model for realistic estimation of the WSN lifetime. An advanced energy consumption model should consider the path loss, the modulation scheme, the hardware used, the coding scheme and so on. While the modulation scheme, the hardware and the coding scheme are specified in the NODEML model, the path loss, as shown in the next section, has been defined according to the environment and its obstacles. NODEML, ENVML and path loss definitions are used to generate low level settings and scripts that can provide a fine estimate of the energy required to transmit a bit over the physical channel.

4.3.3.1 The path loss

In sensor networks, path loss can play a crucial role since avoiding the path loss may cause overestimation of WSN lifetime. The optimistic evaluation of resources is particularly dangerous for WSNs since the resources come with significant restrictions especially in terms of energy. The path loss is reduction in transmitted signal strength as a function of distance, which determines how far apart two sen-

sor devices can be and have reliable communication between the devices [Pahlavan and Krishnamurthy, 2009]. The core of signal coverage calculations for any environment is a path loss model, which relates the loss of signal strength to the distance between two terminals and the operating frequency.

Indoor radio propagation is dominated by the same mechanisms as outdoor propagation: reflection, scattering, diffraction, refraction, absorption and depolarization. However, conditions are much more variable. The indoor environment differs widely due to the increased number of obstacles, layout of rooms, presence of multiple walls and floors, windows and open spaces. Altogether these factors have a significant impact on path loss in an indoor environment. Due to the irregularity in the position of obstacles and layout of the rooms, the channel varies significantly with the environment making the indoor propagation modelling relatively inconsistent and challenging especially for modelling. The propagation and path loss models are usually based on empirical studies on the system considered.

Accurate modelling of the actual environment is very complex as the communication systems operate in complex propagation environments. In practice, most of the simulation studies make use of the empirical models that have been developed based on empirical measurements over a given distance in a given operational frequency range and a particular environment [Rappaport, 1996]. Some of the most common empirical models include Okumura Model, Hata Model, COST 231-Walfish-Ikegami Model, Erceg Model, ITU Indoor Path Loss Model, Log-Distance Path Loss Model etc. [Rappaport, 1996; Seybold, 2005]. When considering an indoor propagation environment for path loss, the material used for walls and floors, the layout of rooms, windows and open areas, location and materials obstructing etc. should be taken into account, as all of these factors have a substantial impact on the path loss in an indoor environment. The complexity of signal propagation in the indoor environment makes it difficult to obtain a single model that illustrates path loss across a wide range of environments. The following is a commonly used simplified model for path loss as a function of distance [Goldsmith, 2005].

$$P_r\,(dBm) = P_t\,(dBm) + K(dB) - 10\gamma\, log_{10}\left(\frac{d}{d_0}\right) \tag{4.1}$$

where, K is the path loss factor, which depends on antenna characteristics

and the average channel attenuation. γ is the path loss exponent, d_0 is reference distance for the antenna far field, and is typically assumed to be 1-10 m for indoor scenarios and 10-100 m for outdoor scenarios. These values can be obtained to approximate either an analytical or empirical model. In particular, the free space path loss model, two-ray model, Hata model, and the COST extension to the Hata model are all of the same. P_r is the received signal strength and P_t is the transmitted signal strength. K can be calculated as:

$$K\ (dB)\ =\ 20\ log_{10}\frac{\lambda}{4\pi d_0} \tag{4.2}$$

The value of λ depends on the propagation environment. This path loss model is specified in the ENVML model, and together with the physical environmental model is used in order to define the path loss between any two nodes. Please note that, the existing simulation packages, and modelling architectures do not consider the effects of path loss to best of our knowledge. In this approach, the value of γ is fixed at 2 for free space and the losses are introduced for each partition (obstacle) that is encountered by a straight line connecting the receiver and the transmitter. Please refer to Table 5.1 for the decibel loss values measured for different type of partitions, at 2.4 GHz [Pahlavan and Krishnamurthy, 2009]. In order to add the effects of obstacles between the transmitter and the receiver, the fixed path losses per existing obstacles is added to the free space path loss.

4.4 PlaceLife implementation

4.4.1 The pervasive computing-based health care system: case study

Pervasive computing based health care systems are considered as case study for demonstration of the new architecture. Numerical results are presented in turn. Static nodes and small distance indoor communication is assumed which is commonly used for these applications of WSNs.

Recent technological advancements in WSNs have opened up new prospects for a variety of applications, including healthcare systems [Alwan et al., 2006; Lorincz

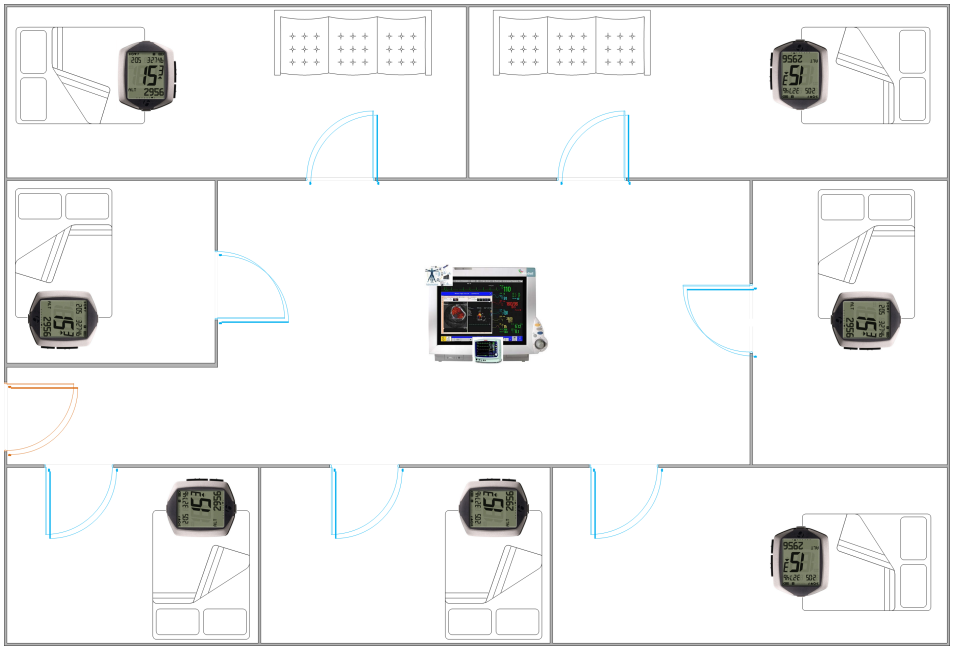

Figure 4.3: Hospital scenario considered

et al., 2004; Shnayder et al., 2005]. WSN implementations on pervasive computing based health care systems avoid various limitations and drawbacks associated with the wired sensors providing a better-quality of care, quicker diagnosis, more intense collection of information (can be employed for statistical analysis) and at the same time keeping the cost and resource utilisation to minimal. Monitoring facilities introduced by using WSNs are particularly useful for early detection and diagnosis of emergency conditions, as well as keeping track of the diseases for patients. WSN based health care systems are also useful for providing a variety of health related services for people with various degrees of cognitive and physical disabilities [Alemdar and Ersoy, 2010]. In [Alwan et al., 2006], the authors present a pilot study conducted in collaboration with the volunteers of America National Services. A number of In-home Monitoring Systems (IMS) were deployed in an assisted living setting. Similarly “CodeBlue", introduced in [Malan et al., 2004], is a wireless sensor infrastructure for seamless transfer of data among caregivers, and efficient allocation of hospital resources. The authors in [Pallikonda Rajasekaran et al., 2010] aim to increase the availability of medical care in order to reduce the demands on hospital services. At the same time the WSN system implemented improves the long-term care and recovery of patients.

In the context described above, the case study (graphically illustrated in Figure 4.3) represents the concept of an in-hospital WSN that allows the personnel of the hospital to monitor patients' vital sign data with the help of pulse-oximeters. The monitoring system consists of two types of nodes: a monitoring station and seven oximeter nodes, forming a star-network. Each pulse-oximeter monitors the patient continuously and a measurement is sent to the monitoring station every three seconds. In case the oximeter reads a value other than a defined threshold, an alert message is sent to the monitoring system, and the system goes into a *warning* mode in which sensor readings are sent to the monitoring station more frequently (i.e., once every 200 milliseconds), hence facilitating continuous monitoring of patients and allowing real-time responses in case of emergency conditions.

When WSNs are employed, small-size network nodes operating at low power and cost, provides easy integration in medical applications. Such systems automate patient monitoring to increase the quality of care in clinical environment and in disaster scenes. The advantages of a WSN are plentiful for smart health care, such as:

- Continuous and real time monitoring: Patients can be continuously monitored, allowing real time responses for emergency conditions.
- Portability: Sensor nodes are very small and operate with minimal patient input, facilitating the ease of monitoring.
- Ease of deployment, reconfiguration, and scalability of the nodes.

WSNs carry the promise of enhancing and boosting the quality of care drastically over a wide variety of settings. Sensor applications such as image sensing, blood pressure monitoring, glucose monitors, thermometers, electrocardiography etc are widely being used in modern medicine. With the advancements in the information technology, medical sensors have become increasingly interconnected with other devices. The sensor nodes are usually equipped with low-power radios such as IEEE 802.15.4. The radio has communication range up to few tens of meters and can transmit at the rates of 10 kb/s to 250 kb/s. [Ko et al., 2010]. Although various data link layer access methods can be used, the Timeout MAC

(T-MAC) has been chosen in this case study. T-MAC [Ye et al., 2004b] is a contention based MAC protocol that uses synchronised sleep schedules between the nodes in a WSN to conserve energy. Also T-MAC provides both collision avoidance and reliable transmission with retransmissions. When pervasive computing based health care systems are considered for evaluation and optimisation, since the area of application is closely related with human life, the overall process should be considered in most realistic way. Also since the lifetime of the sensor nodes is the main limitation, it is essential to use the most optimum design, and the most realistic evaluation settings. Therefore an evaluation tool which incorporates path loss in indoor communications is required.

4.4.2 PlaceLife applied to the wireless health monitoring system

In order to show the effectiveness of the architectural approach and the PlaceLife plugin, a case study is presented about a wireless health monitoring system. The numerical results are also presented to show the effects of realistic simulation scenarios, where environmental factors are taken into account.

The system considered monitors vital signals in a hospital environment. Wireless networked sensors enable dense spatio-temporal sampling in spaces, ranging from personal to physical environment [Alemdar and Ersoy, 2010]. In this approach, the feasibility of continuously monitoring heart rate and saturation of patients haemoglobin is considered.

Figure 4.3 represents the concept of an in-hospital WSN that can be used to monitor patient vital sign data with the help of the pulse-oximeter. The monitoring system consists of two types of nodes: a monitoring station and seven oximeter nodes, forming a star-network. A CC2420 chip, compatible with 802.15.4, is used to provide wireless communication, operating at 2.4 GHz and providing a data rate of 250 kbps. It employs Direct Sequence Spread Spectrum (DSSS) modulation in combination with Offset - Quadrature Phase Shift Keying (O-QPSK) modulation. The oximeter node operates at 4.8 mW [Chipara et al.; Tavakoli et al., 2010]. The pulse-oximeter monitors the patient continuously and a measurement is sent to the monitoring station once in every second. In case the oximeter reads a

value other than the defined threshold, an alert message is sent to the monitoring system. Hence, facilitating continuous monitoring of patients and allowing real time responses in case of an emergency.

4.4.2.1 Numerical Results

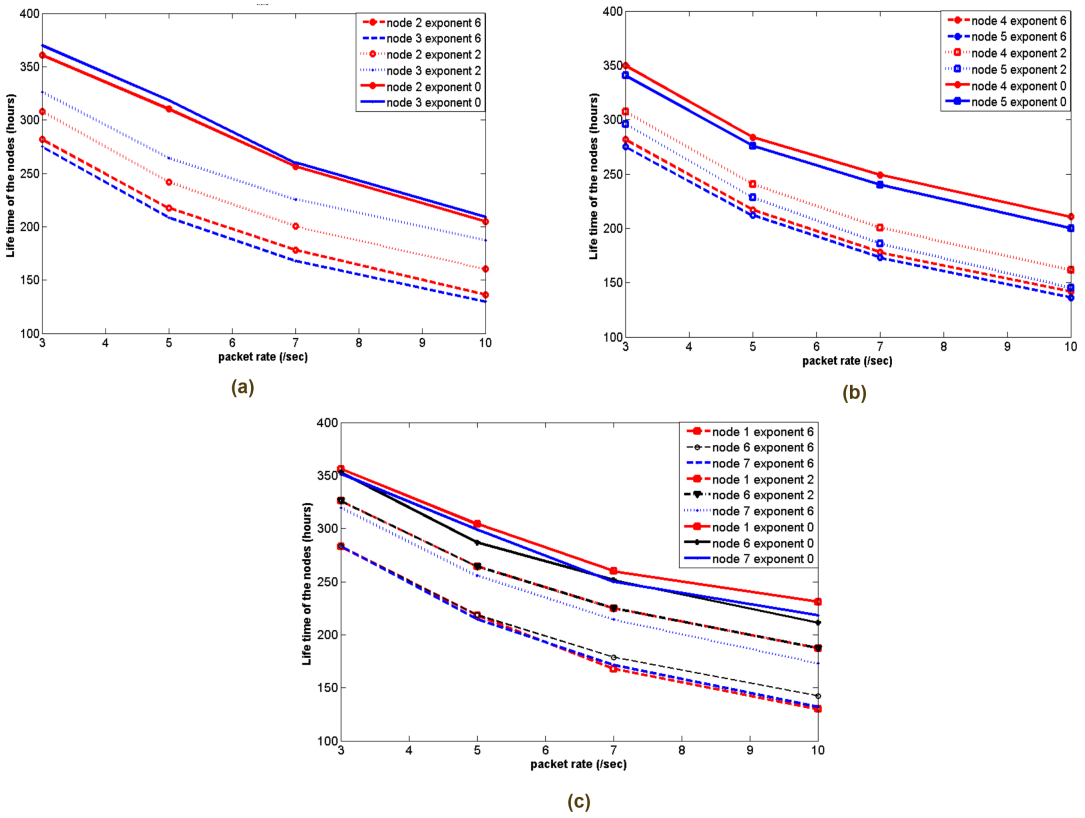

Figure 4.4: Life time of the nodes

The lifetime of the nodes considered is presented as a function of number of packets per minute in figure 4.4 for obstacles with exponent (γ) two, and six. Please note that exponent two can be used for wood and six is mainly for metal obstacles. The obstacles considered are mainly the walls used for indoor segmentation in Figure 4.3. Numerical results are also presented for a scenario where the path loss caused by the obstacles is avoided (exponent is zero). The results clearly show that avoiding path loss would cause overestimation of the WSN lifetime. More

precisely, for node six the life time is 370 hours when the obstacle is metal, 400 hours when the obstacle is wood and 520 when the path loss is avoided. In other words the resources can be overestimated up to 19.5 % if the path loss factor is avoided. PlaceLife's ENVML model allows the users to incorporate various path loss models (in this study he one in [Pahlavan and Krishnamurthy, 2009] is used) for the estimation of the WSN lifetime. The MAC layer dictates the states of the radio. Power consumption is based on the time the radio is on (either listening, transmitting, or receiving) i.e. the state of the radio and how long it stays in each of the states. Only hardware components consume energy: The Radio, the processor, other electronics, the sensors. In order to calculate the life time of each of the nodes, the resource manager module (in Castalia)is used, which keeps track of the energy spent by the node and also holds some node-specific quantities such as the clock drift and the baseline power consumption (the minimum power that the node consumes). There are some provisions to track memory or to define different power consumptions for the processor but they are not fully operational. Modules that model hardware devices (i.e, the radio and the sensor manager) send messages to the resource manager in order to signal how much power they currently draw. The resource manager then has a complete view of the total power drawn and based on this it calculates energy consumed each time we have a change in power or periodically (if a power change has not happened for some time). The periodic energy consumption calculation is done based on the set default value. 18720 Joules is the typical energy of two AA batteries. Energy is linearly subtracted based on overall power drawn and time passed, however based on the power-drawn messages modelling, the user can have more advanced energy consumption models, that take into account the history of power drawn to determine the energy left in battery cells.

It is clear that the energy consumption of all the nodes is significantly higher when the partition is metal. When the partition is wood, the life time of the node is approximately 5-10 % more than that of the nodes with metal as the partition. The life time of a node can decrease from 400 hours, down to 370 hours, when the exponent of the obstacle increases. In other words, as the exponent of the obstacles increases, the effects of path loss also increase since the amount of retransmissions lead to higher energy consumptions. Numerical results presented in Figure 4.4

show the effects of potential overestimation of resources in case the obstacles of a specific scenario are avoided. The PlaceLife plugin allows engineers to consider the nature of the obstacles of the environment in details, thus providing a more realistic performance measurement. While various components of PlaceLife are employed, the design of the simulation does not get complicated since the architecture presented is user friendly. Please note that the multi-view architectural approach allows the user to isolate the physical environment an incorporate various factors such as path loss, shadowing etc.

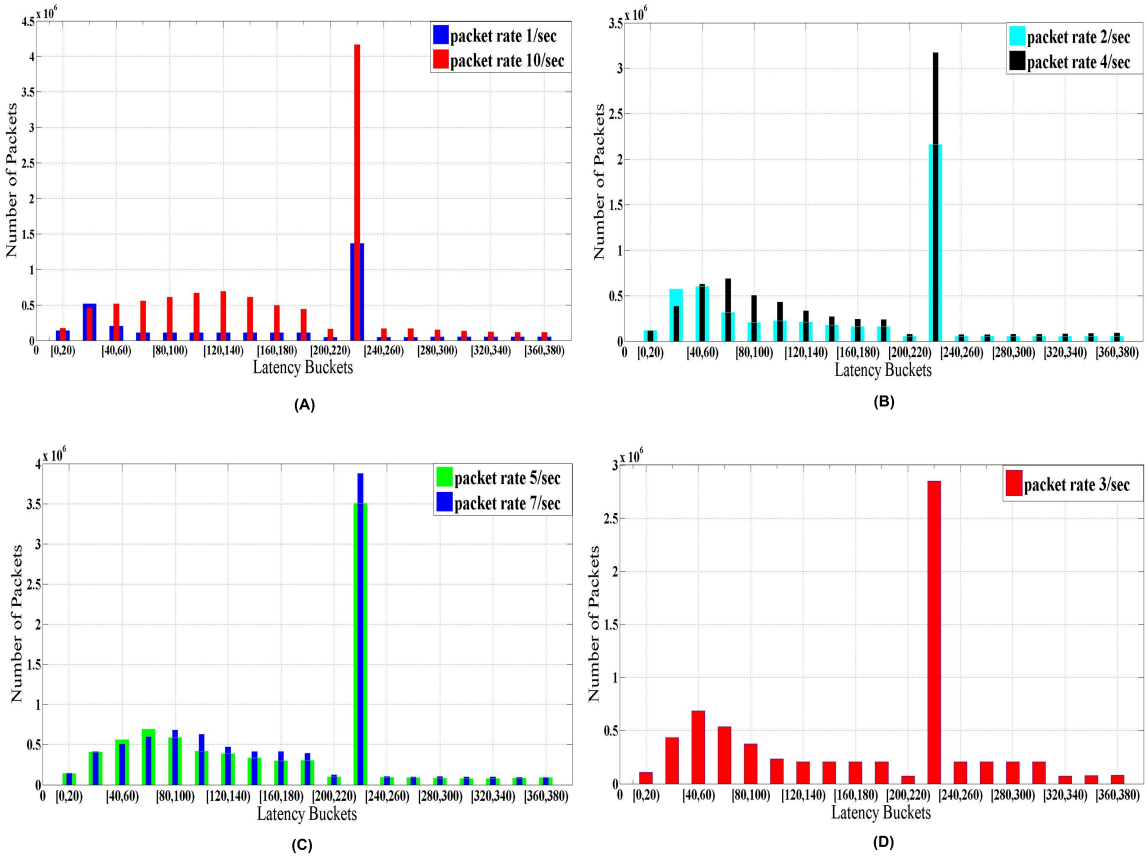

Figure 4.5: Latency of the nodes

The simulation tool employed allows the user to consider other measurements in addition to the life time of the WSN nodes. Various performance measures such as response time (latency), the number of dropped packets etc. can be analysed in detail. Figure 4.5 shows the latency of the packets received by the monitoring

station. The packet rate sets the rate at which packets are transmitted by each node in packets/sec. The start-up delay sets a delay in seconds than the application will start transmitting after the node has been activated. In the case study considered, the start-up delay is set to 0. The latency histogram parameters set how the application level latency is to be recorded. Listen interval is the time the node stays on listening each cycle. Knowing the duty cycle one can then define the time the node sleeps each cycle. After we had one listen and one sleeping interval, the cycle starts again. It is good to have the listen interval to be small so the sleeping interval for a given duty cycle is small too thus latency in data delivery is minimized. Castalia results parses the file and finds out what output is recorded by the different modules. Application produces two kinds of output relating to packet latency and packets received. The latency output is a histogram with 10 time buckets. The results show that most of the packets have a latency interval of 230 ms. As the packet rate increases, the number of packets within the same latency interval also increases. The figure also demonstrates the flexibility of the considered system in terms of performance, availability, and energy-related measures.

4.4.3 PlaceLife applied to the home automation system: Numerical Results and Discussions

In order to show the effectiveness of the architectural approach and the PlaceLife plugin, further in this section we consider a case study about a home automation system, and we present numerical results of its simulation. The numerical results also show the effects of realistic simulation scenarios, where environmental factors are taken into account.

4.4.4 Home automation - Temperature Control and Fire alarm system

Monitoring and automatic control of building environment is a case study considered quite often [Gill et al., 2009; Han and Lim, 2010]. Home automation can include the following functionalities: (i) heating, ventilation, and air condition-

ing (HVAC) systems; (ii) emergency control systems (fire alarms); (iii) centralised control lighting; and (iv) other systems, to provide comfort, energy efficiency and security. In order to validate our approach we consider the fire alarm system and the automatic heating application.

The fire alarm system is composed of temperature sensors, smoke detectors and sprinkler actuators. In our fire alarm implementation we assume that all the temperature sensors monitor the temperature at regular intervals Δt_1. When a temperature sensor reads a value that exceeds a specified threshold T and a smoke sensor detects smoke all the sprinklers are activated. The value Δt_1 and the threshold T are assumed to be 30 seconds and 50 Celsius degree, respectively. The automatic heating application is composed of different temperature sensors, a base station and various heaters. In our automatic heating application the temperature sensors send readings at regular intervals Δt_2 to the base station. This is placed at the center forming a star topology. The base station averages the readings and decides whether or not the central heating system should be on. More specifically the base station works in the following way:

- if the heating is turned on and the average temperature is greater than the maximum temperature T_{max}, the central heating system turns off.
- if the average temperature is less than the minimum temperature T_{min}, the central heating system turns on.

The value Δt_2 is set to be 30 seconds while $T_{min} = T_{max} = 22$ Celsius degree.

We assume the fire alarm system and the automatic heating application are deployed in a building composed of three floors. Each floor has the same floor plan that is shown in Figure 4.6.

Figure 4.6 represents the floor plan of the apartment, containing the temperature and smoke detector nodes. Various obstacles are used in the case study, to study the effect of these obstacles on the performance of the various nodes used. Wooden partition, glass partition, brick walls, etc. are used. For the sake of representation, we use numbers to represent sensor nodes monitoring temperature and smoke. Nodes 1, 2 represent the temperature and smoke detector nodes respectively in the bathroom area. Nodes 3,4 respectively represent the temperature

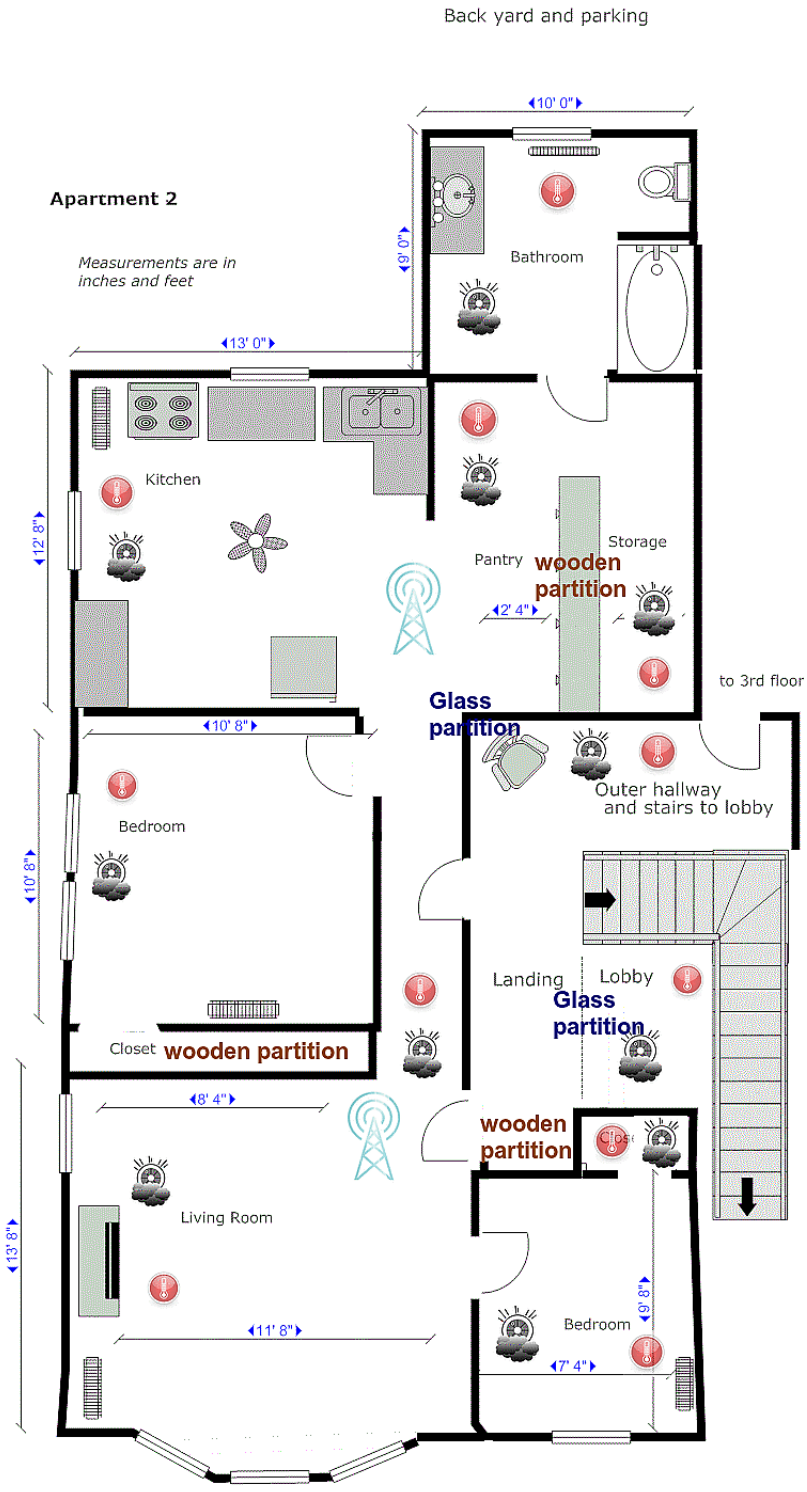

Figure 4.6: Home automation - case study

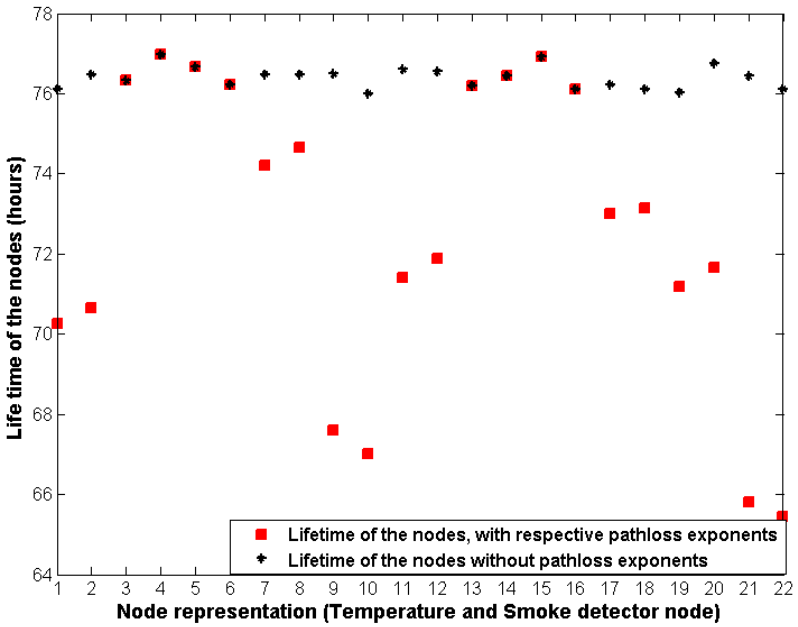

Figure 4.7: Life time of the nodes

and smoke detector nodes in the pantry. Nodes 5, 6 represent the temperature and smoke detector nodes respectively in the kitchen, nodes 7,8 represent temperature and smoke detectors respectively in the storage area and nodes 9, 10 represent the nodes from the outer hallway and stairs to the lobby. All these temperature nodes in these areas sense and send to the base station located close to the pantry.

In the larger bedroom, nodes 11, 12 represent the temperature and smoke detectors nodes respectively, nodes 15, 16 represent temperature and smoke detector nodes in the living room, while 17, 18 represent temperature and smoke detector nodes in the smaller bedroom. In the closet attached to the smaller bedroom, nodes 19, 20 represent temperature and smoke detector nodes respectively. Nodes 21 and 22 respectively represent temperature and smoke detector nodes in the lobby. Nodes 13, 14 are temperature and smoke detector nodes respectively, located close to the base station in the living room. All the temperature nodes in these areas sense and send their data to the base station in the living room. The base stations are connected to each other using peer-to-peer connection.

Figure 4.7 shows the energy consumption of each node in a free space environment and when the path loss due to obstacles (mainly due to partitions) is

introduced. It is evident that ignoring the effect of path loss would be an optimistic assumption when energy consumed by each node is considered. The obstacles considered are mainly the wall partitions used for indoor segmentation. The results clearly show that avoiding path loss would cause overestimation of the WSN lifetime. More precisely, the lifetime of the nodes 1 and 2 deployed in the bathroom area is about 76 hours with no path-loss as compared to 70.5 when the exponent due to the brick wall separating the bathroom and the pantry are considered. Similarly, the lifetime of the nodes 21 and 22 is about 65.5 hours when the attenuation due to the glass partition in the lobby area and also the brick wall separating the landing area and the living room, as compared to 76 hours ignoring the effects of path loss. It can also be observed that the nodes 3, 4, 5, 6, 13, 14, 15, 16 are not affected by path-loss as they are not enclosed by walls or any obstacles. Hence, their lifetime is roughly about 76.5 hours.

It is evident that ignoring the effect of path loss would be an optimistic assumption when energy consumed by each node is considered. Results presented in Figure 4.7 are particularly important to show the usefulness of a detailed and a realistic modelling tool. Our PlaceLife plug-in allows engineers to consider the nature of the obstacles of the environment in details, thus providing a more realistic performance measurement.

4.4.5 Summary and Final Remarks

In this chapter, a rich modelling environment has been proposed, supported by a powerful programming framework, for the model-driven engineering of wireless sensor networks. The modelling viewpoints and conceptual elements have been carefully designed in collaboration with colleagues from various domains, such as software engineering, wireless sensor networks, and telecommunications. The programming framework functioning has been tested by realizing a plug-in devoted to energy-related simulation of WSNs. Two case studies have been presented to show the effectiveness of the framework and the plug-in.

The main novelty of this approach with respect to all the above mentioned literature is the clear separation of concerns amongst the software architecture of the application, the hardware configuration and the WSN deployment topology.

This promotes **reuse** of models across projects and organisations. The modelling language for the physical environment (e.g., ENVML) allows engineers to easily define the physical environment in which the WSN will be deployed and to assess the **deployment** of the WSN nodes in their actual physical environment ealier in the development life cycle. Finally, another important point that sets apart A4WSN from the other approaches is its **extensibility**. Basically, it allows third-party researchers and developers to reuse the A4WSN modelling environment and programming framework when developing new analysis and code generation engines. In this context, third-party researchers can focus exclusively on solving their peculiar issues, while spending minimal effort and implementation time on realizing the facilities already provided by A4WSN out of the box.

Chapter 5

Simulation for WSNs

5.1 Introduction

Energy consumption of nodes is a crucial factor that constrains the networks life time for WSNs. The main concern in the existing architectural and optimisation studies is to prolong the network lifetime. The lifetime of the sensor nodes is affected by various components such as the microprocessor, the sensing module and the wireless transmitter/receiver. The existing works mainly consider these components to decide on best deployment, topology, protocols and so on. Recent studies have also considered the monitoring and evaluation of the path loss caused by environmental factors. Path loss is always considered in isolation from the higher layers such as application and network. It is necessary to combine path loss computations used in physical layer, with information from upper layers such as application layer for a more realistic evaluation. Simulation is an inevitable methodology for specification, design and analysis of computer and communication networks. It is extensively used at all levels ranging from hardware to network. However, to obtain valid results that properly predict the behaviour of a real system, all relevant effects must be captured in the simulation model. This task has become very challenging today, since one can rarely model and simulate the different levels in isolation. Rather, advanced system optimization causes dependencies in the behaviour across all layers. Further, systems have become extremely complex and so have the simulation models. To get relevant statistical results and to

cover critical corner cases the simulated time (i.e., the time elapsed in the simulated system) has to be sufficiently long. To avoid excessive simulation time even on high-performance computers, modelling not only has to be proper but also efficient.In this chapter, a simulation-based case study is presented that uses path loss model and application layer information in order to predict the network lifetime. Physical environment is considered as well. This work shows that when path loss is introduced, increasing the transmission power is needed to reduce the amount of packets lost. This presents a trade-off between the residual energy and the successful transmission rate when more realistic settings are employed for simulation. It is a challenging task to optimise the transmission power of WSNs, in presence of path loss, because although increasing the transmission power reduces the residual energy, it also reduces the number of retransmissions required.

5.2 Simulators for WSNs

In this section, an overview of literature on the existing simulators is presented. Simulation tools for WSNs can be classified based on the level of complexity in to three main categories: Instruction, algorithm and packet level simulators. A detailed taxonomy on WSN simulation tools is also presented in [Du et al., 2014; Lahmar et al., 2012]. The unique features of simulation tools in various categories is also presented in detail, along with the features of classical simulators for WSNs.

Instruction level simulators

Instruction level simulators are often regarded as emulators. They model the CPU execution at the level of instructions or even cycles. TOSSIM [Levis et al., 2003], Atemu [Polley et al., 2004], Avrora [Titzer and et al., 2005] are well known emulators. TOSSIM is the most commonly used emulator. However, compared to other emulators, it is not the most precise one. TOSSIM, is a platform specific simulator (a TinyOS mote simulator) which can compile any code written for TinyOS to an executable file. TinyViz, is the basic GUI for TOSSIM which can visualize and interact with the running simulations. TOSSIM is specific for TinyOS applications on Mica motes sensors and do not include power models. Avora, is a java-based

emulator used for programs specifically written for AVR microcontrollers produced by Amtel and the Mica2 sensor modes. Atemu provides low-level emulation of the operation of individual sensor nodes. A unique feature of Atemu is its ability to simulate a heterogeneous sensor network. It is scalable and its high fidelity platform is used as a pre-deployment tool for sensor networks.

Algorithm level simulators

Shawn [201, 2012], AlgoSensim [Jacques and Marculescu, 2011], and Sinalgo [201, 2011d], are well known algorithm level simulators with emphasis on the logic, data structure and presentation of the algorithms. They rely on some form of graphical data structure to demonstrate the communication between the nodes. Shawn is a very powerful tool in simulating large scale networks with an abstract point of view. It supports distributed protocols and generic high level algorithms. AlgoSensim focuses on network specific analysis of algorithms like localization, distributed routing, and flooding. AlgoSensim mainly facilitates the implementation and quality analysis of new algorithms. Sinalgo focuses on the verification of network algorithms and abstracts from the underlying layers. It also offers a message passing view of the network. Sinalgo can be employed for quick prototyping and verification in freely customizable network settings.

Packet level simulators

OPNET, Qualnet, NS-2, GloMoSim, are some of the most commonly used packet level simulators. They implement the data link and physical layers in the OSI network layers. Hence, radio models, 802.11b or newer MAC protocols, fading, collisions, noise and wave diffractions are commonly implemented. Network Simulator (NS) is a discrete event simulator written in combination of C++ and OTcl. OTcl is an object oriented scripting language, developed mainly for networking research. It provides extensive support for simulation of TCP, multicast protocols, and routing for wired and wireless networks. With protocol implementations being widely produced and developed, the extensibility of NS-2 has been a major contributor to its success. It has an object-oriented design which allows for easy creation of new protocols. The key features for WSNs include battery models,

hybrid simulation support, sensor channels, scenario generation tools and a visualization tool [Issariyakul and Hossain, 2008]. Scalability, lack of application model and the lack of customization are few limitations of NS-2 along with lacking an application model [Chen et al.]. OPNET [201, 2011a] and Qualnet [201, 2011c] are commercialized network simulator software with powerful standard modules and they provide good simulation environment. OPNET is an excellent choice to simulate Zigbee based networks with the implementation of Zigbee protocol and IEEE 802.15.4 MAC protocol. However, performance measures related with energy are not available in OPNET, which is a major set back, as energy is a very significant parameter for performance evaluation. Qualnet performs well in simulating large scale sensor networks due to its scalability in wireless simulation, but OPNET simulation requires a long time when the number sensors considered is large.

The above mentioned simulators use rather simple radio/channel models [Kotz et al., 2004]. Also, the simulators are still platform specific and moderately scalable, making them unsuitable for protocol/algorithm design and testing. The major power consumption of the node is based on the time the radio is on, either transmitting, receiving or listening, and how long the radio stays in each of the states. Hence, it is also of significant importance to consider the energy consumed for listening as well, for performance evaluation. Furthermore the environmental details and especially the effects of path loss, effect of collisions have not been considered in any of the given simulation packages.

5.3 PlaceLife

Figure 5.1 outlines the main components of this approach. This has been implemented in a tool called PlaceLife. PlaceLife takes advantage of the three modelling views (namely, SAML, NODEML and ENVML) in order to provide an estimate of the WSN lifetime. All modelling views are analysed, combined and translated into low level simulation scripts that can be executed to estimate the WSN lifetime.

An *environment editor* allows the user to specify the physical environment by using a graphical editor. The environment can include different obstacles and different sensors. An obstacle can have different properties such as the material it is made of and its size. These properties are used in path loss computation.

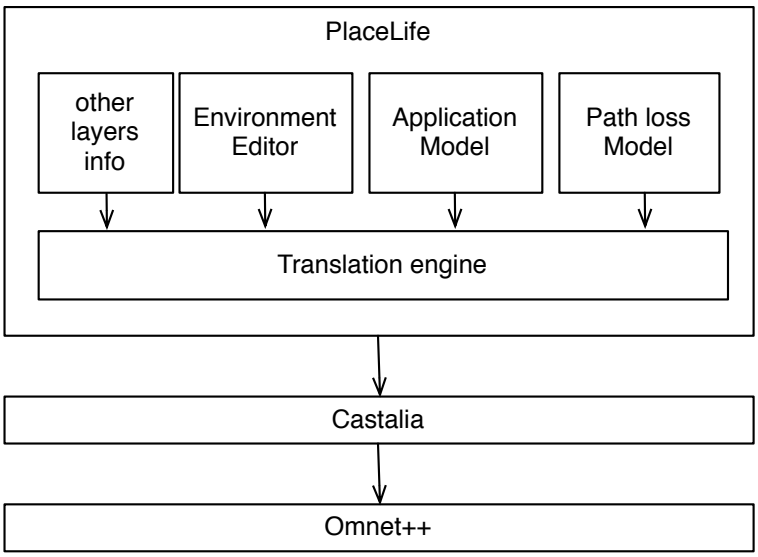

Figure 5.1: PlaceLife

Sensors can be easily placed in the environment.

An *application model* defines the behaviour of nodes. From this model various performance parameters such as transmission and sensing rates can be derived.

PlaceLife considers information from *other layers* such as network, data and physical layers to have a more realistic approximation for the life time. At network layer different protocols such as AODV [Che-Aron et al., 2010] and DSR [Ben Chikha et al., 2011] can be specified but also static routing can be defined. This can be easily specified on the environment model. Although various data link layer access methods can be used, the Timeout MAC (T-MAC) has been chosen in this case study. T-MAC is a contention based MAC protocol that use synchronised sleep schedules between the nodes in a WSN to conserve energy [Ye et al., 2004a]. Also T-MAC provides both collision avoidance and reliable transmission.

5.3.1 Path loss

Path loss is the attenuation in power density of an electromagnetic wave as it propagates. Path loss is consequence of many effects such as free-space loss, refraction, diffraction, reflection, aperture-medium coupling loss, and absorption. Path loss is

Table 5.1: Partition dependent losses for 2.4 Ghz

[Pahlavan and Krishnamurthy, 2009]

Attenuating Material	Signal attenuation in dB
Wood	2
Metal frame, glass wall into building	6
Office wall	6
Metal door in office wall	6
Cinder wall	4
Metal door in brick wall	12.4
Brick wall next to metal door	3

also affected by other factors such as propagation medium (dry or moist air), the distance between the transmitter and the receiver, and the frequency of the signal. When the effects of path loss are not considered, the evaluation of underlying structure can become optimistic, since the problems associated, retransmissions and the way this phenomena affects the energy consumption are not taken into account.

In this approach a *path loss model* can be specified by the user. This model is used together with the physical environmental model in order to define the path loss between two nodes. In this work, the indoor environment and the dependant path loss model [K.Pahlavan and P.Krishnamurthy, 2009] is considered. This is one of the most commonly used path loss models that defines the behaviour of signal strength in an indoor area. The path loss behaviour is dependent on the distance between nodes and the attenuation factor added by the objects. The attenuation can vary based on several factors such as the construction materials (e.g., wood, glass and concrete) and the object size. In the Table 5.1 the attenuation values in dB introduced by various materials is presented. The dependant path loss model can be expressed as [K.Pahlavan and P.Krishnamurthy, 2009]:

$$L_P = L_0 + 20log(d) + \sum m_{type} w_{type}$$

where, L_P represents the path loss between two points, L_0, is the path loss in free space environment, m_{type} refers to the number of objects of the same type and w_{type} is the loss in decibels attributed to that particular object.

5.3.2 Software Architecture Modelling Language (SAML)

The SAML modelling language allows architects to define the software architecture of the WSN application. We formalise the structure of the SAML language and its constructs by defining its underlying metamodel. Figure 5.2 and Figure 5.3 show the parts of SAML metamodel related to its structural and behavioural concepts, respectively[1].

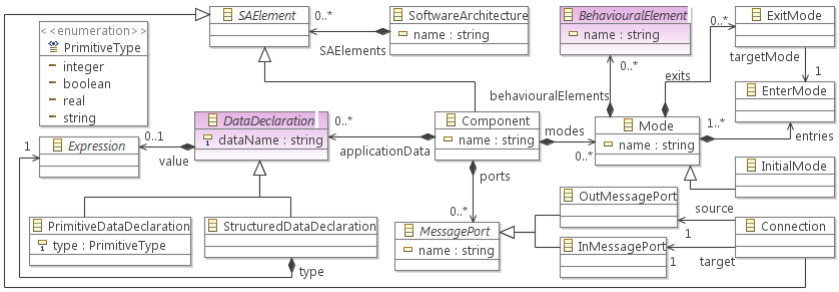

Figure 5.2: SAML Metamodel: structural concepts (external metaclasses in pink)

The **software architecture** of a WSN is defined as a collection of software components and connections. It represents the application layer of the WSN and it is the root element of every SAML model. A **component** is a unit of computation with internal state and well-defined interface

A **mode** represents a specific status of the component. Examples of modes can be, sleeping mode, energy saving mode, etc. Modes are defined at the application layer (in the SAML model) and, while they can be directly related to energy-related modes of a sensor node, they can also be used the represent any logical state of a sensor. At any given time, one and only one mode can be active in a component. The component reacts only to those events which are defined within its currently active mode. An initial mode is the first mode which is active when the component starts up. Mode transitions occur by passing from a special kind of action called **ExitMode** to a special kind of event called **EnterMode** (the concepts of exit and enter mode will be described later in this section). In this way, actions and events can be linked to modes entry and exit points, creating a continuous behavioural flow among modes. Each mode can contain a set of **behavioural elements** thath represent actions, conditions and events which together make up the control flow

[1]The name of abstract classes is shown in italics

within the component from an abstract point of view. In a way, SAML modes may seem similar to UML state machines since they represent the states of a component, and it may switch from a mode to another via a transition; however, in order to manage the complexity of the models, SAML modes can contain only SAML behavioural elements and within a component one and only one mode can be active (no concurrency).

Components interact with other components through **message ports**; they specify the interaction points between a component and its external environment. Input message ports are used to receive incoming messages, while output message ports are used to send outgoing messages. Communication happens by message passing, which means that, a component can send messages from one of its output message ports to input ports of other components. The actual communication method of a message (i.e.,broadcast, multicast or unicast) is specified in the send message action described later in this section. In this context, a **connection** represents unidirectional communication channel between two message ports of two different components. The data contained in a message is accessible by specific actions and events defined in the behaviour of the involved components (see the **SendMessage** action and **ReceiveMessage** event in Figure 5.3).

As previously considered, in SAML each component can contain its corresponding behavioural description. Figure 5.3 shows the part of SAML metamodel related to behavioural concepts. Each component can contain a description of its behaviour in terms of events, conditions and actions. **DataDeclaration** and **BehaviouralElement** correspond to the metaclasses with the same name in Figure 5.2.

An **action** is a special kind of behavioural element which represent an atomic task that can be performed by the component. An action can be performed in response to an event trigger, or because a previous action in the behavioural flow has been executed. Examples of action include: start or stop of a timer, send a message via a specific message port (either as unicast, multicast, broadcast, or scoped), get data from a sensor, call an external service, start a timer, and so on. It is important to describe a new kind of action we introduced called *scoped send message*; basically, this action tells that the set of nodes receiving the message is computed at run-time, depending on the value of a boolean expression; only the

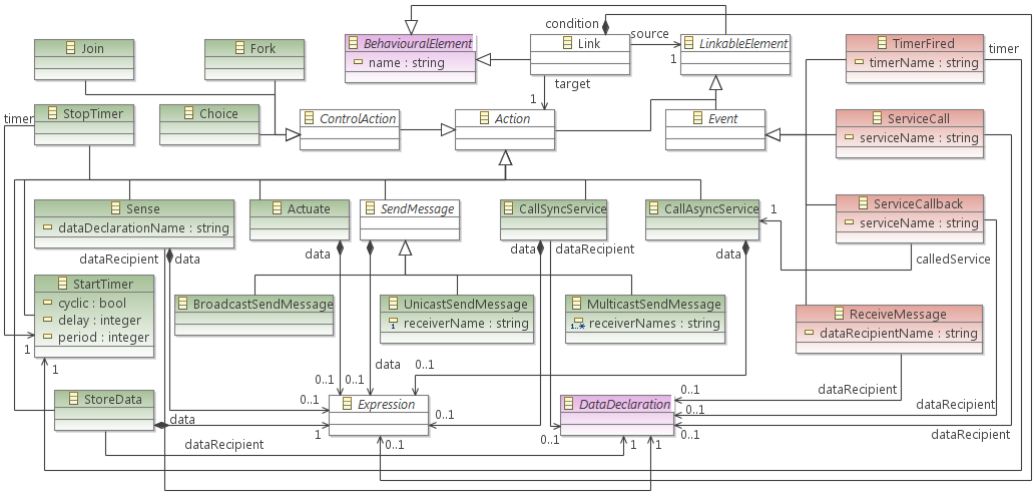

Figure 5.3: SAML Metamodel: behavioural concepts (actions in green, events in red)

nodes whose application data values satisfy the boolean expression will receive the specific message, thus enabling dynamic scope-based interactions within the WSN

An **event** is an SAML behavioural element representing an occurrence that can happen during the execution of a component. An event is triggered in response to either an external stimulus of the component (e.g., the message reception on a input message port), or some internal mechanism of the component (e.g., a timer fired). Examples of event include: entering a specific mode, receiving a message at a given port, an activation of a timer, the receiving of a call from an external service, the receiving of an interrupt from either a sensor or an actuator, etc. A more detailed description of the types of event supported by SAML is provided in the technical report that is accessible on

Events and actions are connected via **links**, they represent the control flow among events and actions. A link helps architects in defining the order in which actions can be executed and the actions that must be executed when an event is triggered. Thus, a link can exist either from an event e to an action a (in this case, a is executed only after e has been triggered), or from an action a to another action b (in this case, b can be executed immediately after a has been executed). Optionally, a condition can be specified in a link; the behavioural flow goes through a link only if its condition evaluates to true. Conditions are defined as boolean expressions which may refer to application data declared in the component, constants, and other operations (see the **Expression** metaclass in

Figure 5.3). SAML exposes five types of expression:

1. **Constant**: a constant representing the value of application data that is considered;
2. **DataRef**: a reference to the value of an application data declaration; it is conceptually similar to a variable reference in Java;
3. **StructureMemberRef**: a reference to a member of a structure; it is conceptually similar to the access to members of a class in Java;
4. **EnumMemberRef**: a reference to a member of an enumeration; it is conceptually similar to the access to a value of a Java enumeration;
5. **Operation**: an application of some kind of operation. Available operations are:
   - arithmetic operations: sum, subtraction, multiplication, division;
   - boolean operations: logical *and*, logical *or*, logical *not*;
   - relational operations: $>$, $\geq$, $<$, $\leq$, $=$, $\neq$.
   - string operations: *length*, *contains*, *substring*, *concat*, *startsWith*, *endsWith*;
   - byte array operations: *hash*, *contains*, *indexOf*, *concat*, *subtract*.

For the sake of brevity, and since the various expressions available in SAML follow classical mathematical, logical, and programming expressions, we do not graphically show them in Figure 5.3, and we do not go into the details of their semantics.

5.3.3 Node Modelling Language (NODEML)

NODEML is our language for describing the low-level details of each *type of node* that can be used within a WSN. A NODEML model contains exclusively low-level, node-specific information. Different WSN applications can reuse the same NODEML models and organize them differently, depending on the requirements of the application. Figure 5.4 shows an overview of the metamodel underlying

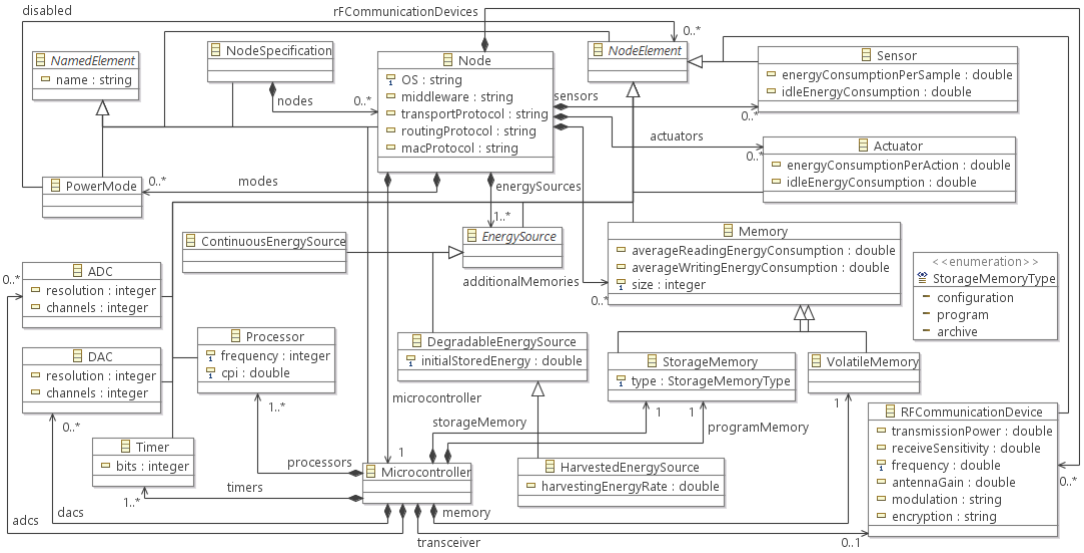

Figure 5.4: NODEML Metamodel

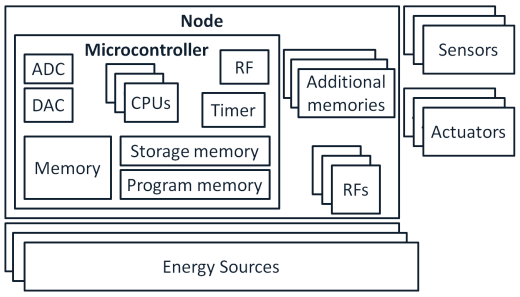

Figure 5.5: an abstract view of the low-level parts of a WSN node.

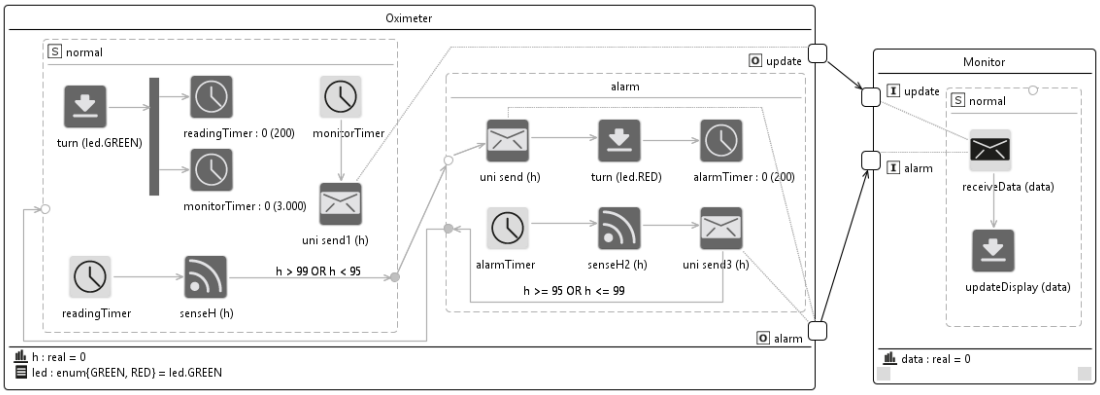

Figure 5.6: Software architecture of the hospital scenario WSN

our NODEML language. In the following we provide a description of the main concepts defined in the NODEML metamodel. For the sake of brevity, we do not go into the details of each element of the language, the details are presented in the the technical report that is accessible on

A **node specification** is the root element of a NODEML model. It is a container of instances of the *Node* metaclass. A node represents a specific node type that can be used within the WSN. A node can have a name (inherited from the *NamedElement* metaclass), its operating system (e.g., TinyOS, Contiki, Mantis, LiteOS), *middleware* (such as TeenyLIME, MiLAN, RUNES

From a structural point of view, according to the NODEML metamodel a WSN node is composed of a set of node elements; in this part of NODEML, we took inspiration from the abstract view of the low-level parts in a typical WSN node as described by Picco and Mottola in

An **energy source** represents any equipment or device used to provide electrical energy to the node. NODEML supports three types of energy sources, namely:

- **ContinuosEnergySource** that potentially never runs out. A classical example of this kind of energy source is AC/DC supply.
- **DegradableEnergySource** that can terminate at any time. As an example, batteries can be represented by a degradable energy source in NODEML.
- **HarvestedSource** that can terminate at any time and harvest additional energy in some way. As an example, in NODEML batteries coupled with a solar panel can be represented as an harvested energy source.

In NODEML, **sensors** are the hardware component performing the actual readings from the environment. Intuitively, a sensor can be seen as a unit that measures a physical quantity and converts it into a signal which can be analyzed and manipulated by a microcontroller. A WSN node can be equipped with zero or more sensors. Today many types of sensors exist, each of them tailored to acquire specific data from the environment; for example, a sensor can get information about the lighting conditions of the environment, its current temperature, presence of smoke or gas, the geographical position of the node, and so on

An **actuator** is the hardware component that can physically operate on the environment. Conceptually, it performs the inverse operation of a sensor, i.e., sensors acquire information from the environment and allow the microcontroller to perform some computation with it, whereas actuators are triggered by some computation of the microcontroller and then perform an action in the environment. A WSN node can be equipped with zero or more actuators. Examples of actuators include: water sprinklers, lights, electronic door locks, motors, airscrews, etc.

An **RF communication device** is a radio device to communicate with other WSN nodes. For example the ChipCon 2420[1]. Technically, in NODEML an RF communication device represents an RF transceiver that can operate on specific bands, such as the 2.4 GHz ISM, XX etc., and are compliant with some IEEE standard which specifies their physical layer and media access control, such as the IEEE 802.15.4

A WSN node commonly requires **memory** as well. NODEML supports two kinds of memory:

- **Volatile memory** that represents the classical volatile memory in computer systems. When power supply is interrupted the stored memory is lost. Usually, the size of a volatile memory in a WSN node ranges from 2Kb to 512Kb

  A **micro-controller** is an electronic device integrated into a single chip, it is commonly used in embedded systems. Examples of micro-controllers are: ATMega128, Texas Instruments MSP430, etc. According to NODEML, a micro-controller can contain one or more processors, zero or more ADCs (abbreviation for Analog-to-Digital Converter) , zero or more DACs (abbreviation for Digital-to-Analog Converter), one or more timers, a volatile memory, a program memory, a storage memory, and an optional radio transceiver.

  The **processor** is the element which physically performs the computational logic of the node. Examples of processors include 8 bit AVR Mega, ARM920T, etc. In NODEML a processor is characterized by its *frequency* (i.e., its clock rate) in MHz, and its cycles per instruction (CPI) representing the number of clock cycles needed for executing a single instruction.

[1] ChipCon 2420 data sheet: http://www.ti.com/product/cc2420

An **ADC** is a device for converting a continuous physical signal into a digital value that "discretizes" it. A **DAC** is a device that performs the inverse operation of an ADC; it converts digital values into continuous physical signals.

Timers are devices apt to periodically trigger the clock of the WSN node. A timer can be implemented either as a hardware or software component and usually works even when the device is in sleep mode, allowing the node to switch from sleep to active power mode. In NODEML, designers can specify the number of bits of a timer (usually they are 16 or 32 bits).

Finally, a node can specify a set of **power modes**, each of them describing a specific configuration of the elements of the node in terms of their power state. Each power mode identifies a set of node elements (such as memory, DAC, RF comm. device, etc.) and identifies which elements are *disabled* (all the other elements of the node are assumed to be active). For example, a given WSN node can have a *Sensing* power mode in which all the radio transceivers are disabled (thus saving all the energy needed to perform networking operations), or a WSN node can have all the sensing devices disabled, and thus focusing only on networking operations, and so on.

Figure 5.7 shows the NODEML model developed for our hospital scenario. Two node configurations have been defined:

- *OximeterNode* is equipped with a *IRProbe* sensor for sensing the percentage of oxygen in the patient's blood and a led actuator for showing the current status of the node to the personnel of the hospital. This node is powered by two AA batteries with up to 18720 Joules and uses a Texas Instruments ChipCon 2420 RF transceiver. The micro-controller used is the low-power Atmel AVR ATmega128 equipped with an ADC for translating the analog values read by the IRProbe sensor into their corresponding digital values. The oximeter node is always active (see the *active* power mode).

- *MonitorStation* has a single actuator device for graphically showing the values received by various oximeter nodes on a digital display. Similar to *OximeterNode*, it uses a Texas Instruments ChipCon 2420

RF transceiver and uses low-power Atmel AVR ATmega128 microcontroller. The monitoring station is always active (see the *active* power mode) and is powered by a classical electrical plug connected to the main electrical system of the hospital. Finally, it is equipped with an additional storage memory for storing a log of all the values received by the oximeter nodes over time.

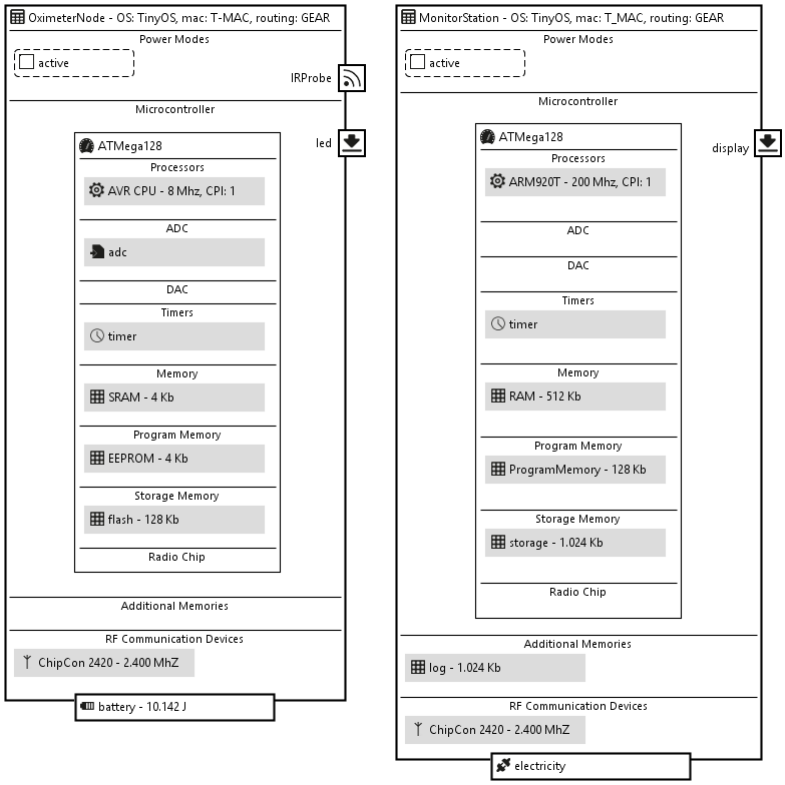

Figure 5.7: Nodes configuration of the hospital scenario WSN

Both nodes use TinyOS[1] as operating system, GEAR as routing protocol, and T-MAC as MAC protocol.

[1] http://www.tinyos.net/

5.3.4 Environment Modelling Language (ENVML)

The ENVML modelling language allows the designers to specify the physical environment in which the WSN nodes are deployed. Figure 5.8 shows the metamodel underlying the ENVML modelling language.

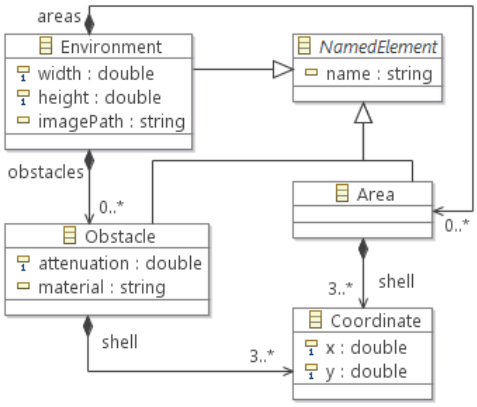

Figure 5.8: ENVML Metamodel

The **Environment** represents the overall area in the 2D space in which the WSN nodes are deployed. The *width* and *height* attributes represent the dimensions in meters of the minimum bounding box of the environment. In geometry, a minimum bounding box is the smallest rectangle that can be drawn around a set of points such that all the points are inside it, or exactly on one of its sides. The four sides of the rectangle are always either vertical or horizontal, parallel to the x or y axis

Any kind of relevant **obstacle** can be placed in the environment. Each obstacle is characterized by the name of the *material* it is made of (e.g., concrete wall, wooden door, glass, etc.), and its *attenuation* coefficient. The latter attribute is a decimal number ranging from zero (when it is totally irrelevant when considering radio signal attenuation, e.g., a sheet of paper), to one (when it totally blocks radio signals, e.g., a panel made of lead). The attenuation coefficient is one of the most important parameters when considering path loss models, network connectivity and coverage, and so on. The shape of the obstacle is given by its *shell*: a sequence of **coordinates** representing the perimeter of the obstacle in the 2D space.

In ENVML an **area** identifies a portion of physical environment in which nodes of the same type can be distributed according to a distribution policy (defined in the DEPML modelling language, see Section 5.3.6). Similar to obstacles, the perimeter of the area is defined by means of its shell.

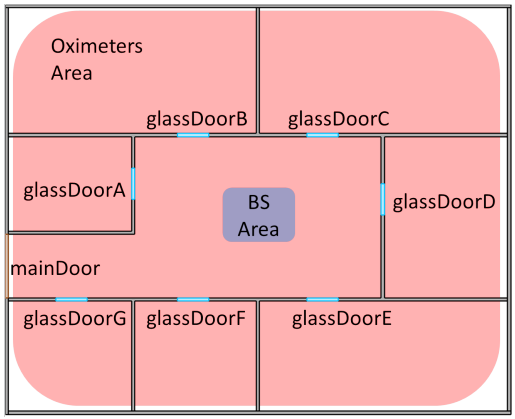

Figure 5.9: Physical environment of the hospital scenario WSN

Figure 5.9 shows the ENVML model representing the physical environment of our hospital scenario. It is a rectangle with 16 and 13 meters of width and height, respectively and it contains three kinds of obstacles that are concrete walls dividing the whole environment into rooms and corridors, a main wooden door on the left, and a glass door for each patients room. Each obstacle is represented by a unique name, its attenuation coefficient and the coordinates of all the points of its perimeter. The physical environment of our hospital scenario contains two main deployment areas:

- *BSArea* is a square area at the center of the environment and will contain the central monitoring station.
- *OximetersArea* its perimeter is the same as the whole physical environment and will contain all the oximeter nodes, one for each patients' room.

It is important to note that the above mentioned solution is one of the possible ones for deployment configurations. Another solution could also

be the creation of a single area for each oximiter. Each oximeter could be placed in the centre of the area. The aforementioned solutions share the same network topology.

5.3.5 Mapping Modelling Language (MAPML)

MAPML is our language for assigning software components to the corresponding hardware node configuration they will be executed on. Fundamentally, a MAPML model semantically represents the classical notion of deployment of software components onto hardware resources

The root of a MAPML model is the **mapping** element that references the linked SAML and NODEML models via the *softwareArchitecture* and *node* containment references, respectively. The **mapping** element is made of a set of **node mappings**, each of them linking a node definition from the NODEML model and a component from the SAML model. The semantics of a node mapping is that the linked component in the SAML model will be physically deployed on the linked node in the NODEML model. A node mapping can contain a set of secondary links, each of them can be seen as a refinement of the node mapping. Secondary links are:

- **SensorMapping** that maps either a *Sense* action or a *SensorInterrupt* event in an SAML component to a *Sensor* device in a NODEML node configuration. Fundamentally, this kind of link allows designers to specify to which physical sensor device does either a sense action or a sensor interrupt event refer to.
- **ActuatorMapping** that maps either an *Actuate* action or an *ActuatorInterrupt* event in an SAML component to an *Actuator* device in a NODEML node configuration. It is similar to the *SensorMapping* concept, but it refers to actuators, rather than sensors.
- **CommunicationDeviceMapping** that maps an SAML message port of the component linked by the parent *NodeMapping* to a NODEML radio transceiver in the node configuration linked by the parent *NodeMap-*

ping. It allows designers to physically map a software port to its corresponding physical radio transceiver.

- **ModeMapping** that maps an energy mode defined in an SAML component to its corresponding power mode in the linked NODEML node configuration. It allows designers to decouple the two concepts of mode we have in SAML and NODEML, and thus it opens for a more flexible definition of modes in the pure "software world", independently from the power modes that the WSN node has in the "physical world".

The MAPML metamodel has some auxiliary metaclasses like **MappingModelRef**, **MappingElementRef** and **MappingLinkEnd** which are used for technical reasons . The interested reader can refer to

For what concerns our hospital scenario, the MAPML model linking the SAML model showed in Figure 5.6 and the NODEML model showed in Figure 5.7 has the following form:

- *NodeMapping_ oximeter* links the *Oximeter* component to the *OximeterNode* node;
  - *ModeMapping_ active* links both the *normal* and *alarm* modes of the *Oximeter* component to the *active* power mode of the *OximeterNode* node. It is important to note that operating modes defined in SAML are pure logical modes, whereas power modes defined in NODEML actually depend on the hardware configuration of the node itself.
  - *SensorMapping_ irProbe* links both the *SenseH(h)* and *SenseH2(h)* SAML sense actions to the hardware *IRProbe* sensor in the NODEML model.
  - *ActuatorMapping_ led*, which is similar to *SensorMapping_ irProbe*, links both the *turn(led.GREEN)* and *turn(led.RED)* SAML actions to the hardware *led* actuator in the NODEML model.

    - *CommunicationDeviceMapping_2420* links the *update* and *alarm* SAML message ports to the *ChipCon2420* RF transceiver defined in the NODEML model.

- *NodeMapping_monitor* links the *Monitor* component to the *MonitorStation* node;
    - *ModeMapping_active* links the *normal* mode of the *Monitor* component to the *active* power mode of the *MonitorStation* node.
    - *ActuatorMapping_display* links the *updateDisplay(data)* actuate SAML action to the hardware *display* actuator in the NODEML model.
    - *CommunicationDeviceMapping_2420* links the *update* and *alarm* SAML message ports of the *Monitor* component to the *ChipCon2420* RF transceiver defined in the NODEML model.

With such a configuration, we now have a clear view of how various elements defined at the software architecture level interact with the hardware. For example, all the communication between the *Oximeter* and *Monitor* SAML components happen between different WSN nodes, whereas all the other actions defined in the control flow are executed locally to the component containing them. Also, the MAPML model establishes which hardware sensor and actuator equipments are actually used for performing the abstract sense and actuate actions defined in the SAML model. This level of flexibility is exactly the main goal of the modelling approach.

5.3.6 Deployment Modelling Language (DEPML)

DEPML is our language for virtually deploying WSN nodes into the physical environment. Figure 5.10 shows an overview of the metamodel underlying the DEPML language.

DEPML allows designers to consider each node configuration defined in a *NODEML* model and to *instantiate* it in a specific area within the physical

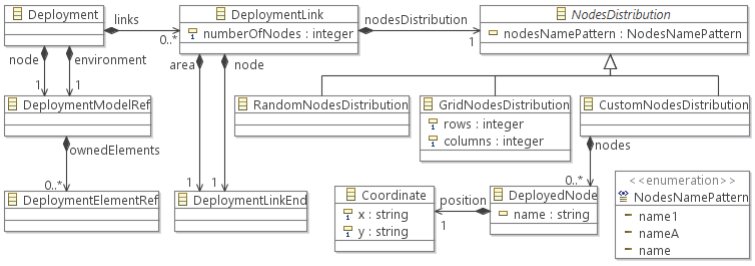

Figure 5.10: DEPML Metamodel

environment defined in a *ENVML* model. A DEPML model contains a single type of link, called **DeploymentLink** linking together a node configuration in NODEML and an area in ENVML. The semantics of the deployment link is that the linked node configuration is instantiated and virtually deployed in the linked area multiple times. This allow designers to focus on generic components and node types in SAML and NODEML, while in DEPML they can reason on the final deployment of the WSN. The number of nodes that are instantiated in the area is defined in the *numberOfNodes* attribute. Within a certain area each node configuration can be **distributed** in three different ways:

- *random*, each node is placed randomly within the area;
- *grid*, nodes are placed on a grid with a certain number of *rows* and *columns*;
- *custom*, each node is manually placed within the area. In this case, each **deployed node** is represented by its *name* (which must be unique within the area) and the coordinates of its *position*.

Also, **nodes name patterns** can be used by designers for declaring the textual pattern of the names of the nodes distributed within the area. They are used as a way to refer to the names used as targets of *Send Message* actions in SAML models. Similar to MAPML, also the DEPML metamodel has some auxiliary metaclasses like **DeploymentModelRef**, **DeploymentElementRef** and **DeploymentLinkEnd**. They are described and discussed in

For what concerns our hospital scenario, the DEPML model is very straightforward. It contains a deployment link between each node defined in the NODEML (see Figure 5.7) and its corresponding area in the ENVML model (see Figure 5.9). More specifically, the DEPML model has the following elements:

- *DeploymentLink_ oximeter* links the *OximeterNode* NODEML node to the *OximetersArea* ENVML area. Since we want to specify that exactly one oximeter node must be deployed in each patient's room, we define a custom nodes ditribution. Thus, we manually define the exact position of the deployed node by means of ten *DeployedNode* elements, each of them containing the coordinates of its position in the environment.
- *DeploymentLink_ monitorStation* links the *MonitorStation* NODEML node to the *BSArea* ENVML area. In this case we specify that the number of deployed node is only one, with a random distribution within the area (we can do this because the area is a square with a side of 0.5 meters, which is exactly the size of the monitoring station node).

The presented DEPML models unveil the flexibility we achieved with the approach. Indeed, if the hospital WSN application can be reused in a different hospital, the SAML, NODEML, and MAPML models can be reused as they are. The only models that must be adapted are the ENVML model for representing the new physical environment with its obstacles and the DEPML model for linking the original NODEML nodes to the new areas, possibly with different values for specifying the number of deployed nodes (e.g., twenty oximeter nodes instead of ten).

In conclusion, all the proposed languages have been designed to provide a good trade-off between genericity, expressivity and accuracy in capturing the various facets of the WSN domain. To this respect, it is fundamental to allow designers to check whether their models are correct with respect to the semantics of the proposed languages. More specifically, allow designers to check whether a model adheres to the structural semantics of its corresponding language (e.g., SAML). supports this feature by leveraging the

well-known notion of **conformance** in Model-Driven Engineering; in other words, in a model m adheres to the structural semantics of its corresponding language (e.g., SAML) if and only if m actually conforms to its metamodel (e.g., the SAML metamodel introduced in Section 5.3.2). Furthermore, in order to ensure a more precise semantics of the languages described in Section 4.2.1, we complemented them with fourteen OCL[1] constraints. For example, in NODEML a constraint ensures that each instance of *Node* must contain at least one program memory, another constraint in DEPML ensures that the coordinates of each manually positioned node must be within the boundaries of the area it is deployed in, and so on. For the sake of readability the description of such constraints are not discussed extensively. If the need for more strict semantics of the proposed languages arises (for instance in order to define WSN applications with specific styles or special configurations), additional OCL constraints can be added to every element of the languages by extending the platform with a suitable plugin. Please, refer to Section 4.2.2 for more details on this feature of the platform.

5.3.7 The translation engine

The *translation engine* takes as an input the environment, application, and path loss models in order to produce simulation scripts. Castalia [201, 2011b] is used as a simulation tool. Castalia is a WSN simulator used for initial testing of protocols and/or algorithms with a realistic node behaviour, wireless channel and radio models. Since Castalia is highly tunable and can simulate a wide range of platforms, it is used to evaluate different platform characteristics. Castalia features an accurate radio model based on the work of the authors in [Zuniga and Krishnamachari, 2009]. It also features physical process model, considering clock drift, sensor energy consumption, CPU energy consumption, sensor bias etc. Unpredictability of the wireless channel, energy spent in transmission/receiving packets, performance degradation experienced by duty cycles, collisions are usually overlooked by simple simulators. However these details are well established in Castalia [Kotz et al.,

[1] Object Constraint Language (OCL) specification: http://www.omg.org/spec/OCL/2.3.1

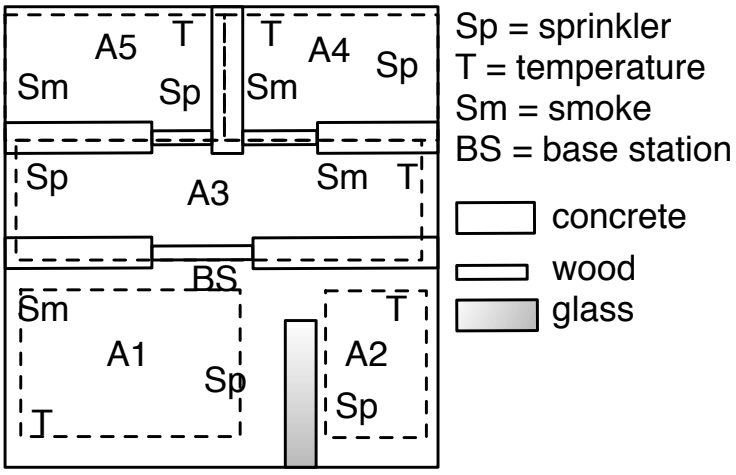

Figure 5.11: Home automation

2004]. All main components that affects the energy consumption of sensor nodes are considered that are the micro-processor, the sensor module, wireless transmitter/receiver and the path loss.

This work emphasises that while Castalia provides a good low level simulation platform; it does not provide any means to specify the application behaviour, the environment and the path loss models. The application behaviour is needed to derive application level simulation parameters. The environment and the path loss models allow the calculation of the path loss. In fact while Castalia assumes that the user provides path loss related parameters, this approach automatically derives those values from high level models such as the environment and path loss.

5.4 Home automation: case study

Monitoring and automatic control of building environment is a case study considered quite often [Gill et al., 2009; Han and Lim, 2010; Kim et al., 2007; Tachwali et al., 2007]. Home automation can include the following functionalities: (i) heating, ventilation, and air conditioning (HVAC) systems; (ii) emergency control systems (fire alarms); (iii) centralized control lighting;

and (iv) other systems, to provide comfort, energy efficiency and security. In order to validate the this approach, alarm system and the automatic heating application is considered as a case study. The fire alarm system is composed of different temperature sensors and smoke detectors that are distributed inside the building. There are also sprinkler actuators used to enable the water flow in case of fire. All the temperature sensors monitor the temperature at regular intervals (every 30 seconds). When a temperature sensor reads a value that exceeds a specified threshold; it sends an alert message to the smoke detector. The smoke detector receives the alert and checks for smoke. An alarm is raised when the smoke is detected. In this case the smoke sensor also activate all the sprinklers.

The automatic heating application is composed of different temperature sensors, a base station and various heaters. The temperature sensors send readings every 30 seconds to the base station. This is placed at the center forming a star topology. The base station averages the readings and decides whether or not the central heating system should be on. More specifically the base station works in the following way:

- if the heating is turned on and the average temperature is greater than the maximum threshold, the central heating system turns off.
- if the average temperature is less than the minimum threshold, the central heating system turns on.

In this work, the scenario of Figure 5.11 is considered. A flat composed of five rooms (A1-A5), with 5 temperature sensing nodes and 5 smoke detector nodes, also considering different obstacles such as wooden doors, walls and glass partition.

5.5 Numerical results and discussions

To understand the behaviour of a particular system it is not sufficient to perform a huge amount of simulations. Simulations must be set up to address the right questions and the results must be analysed thoroughly and be

interpreted properly. In order to show the usefulness and effectiveness of this approach and to analyse various factors affecting the performance in terms of energy consumption of WSNs, the numerical results are presented in this section. The simulation parameters are as follows: CC2420 radio defined by the Texas instruments is used, the output power of the different transmission levels in dBm are varied from 0 to -25dBm. Energy consumption for each transmission level varies. For instance for 0 dBm power consumed for listening (receiving) is 62 mW and for transmission is 57.42 mW. Packet rate is kept at 250 kbps, the radio bandwidth is 20 MHz and the simulation runs for 9000 sec. T-MAC is used as a MAC protocol, and this makes the length of each frame period for all nodes 610 milliseconds, and the duration of listen time out 61 milliseconds.

For this case study, path loss due to the material is calculated and explicitly set the path loss map [K.Pahlavan and P.Krishnamurthy, 2009]. Refer to Figure 5.11 and Table 5.1 [K.Pahlavan and P.Krishnamurthy, 2009] for each type of obstacle and for its contribution to path loss. For the sake of the presentation, the numbers are used to represent sensors. Node 0 represents the base station. Nodes 1,4,5,7, and 9 monitor temperature in areas A1,A5,A4,A3, and A2 respectively. Nodes 2,3,6, and 8 monitor smoke in the areas A1,A5,A4, and A3 respectively. Table 5.2 and Table 5.3 show the energy consumed by the nodes for the application scenario considering the path loss phenomenon and ignoring the path loss respectively. Similarly, Figure 5.12 shows the difference in energy consumed by each node for two different cases. In case one path loss is ignored, and for the next set of results the path loss is present. Comparative results are presented to show the

Table 5.2: Energy consumed by the nodes in joules, considering path loss

nodes	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
energy	100.7	84.43	84.57	89.33	89.49	89.84	88.78	84.42	84.52	86.24

Table 5.3: Energy consumed by the nodes in joules, ignoring path loss

nodes	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
energy	81.41	81.43	82.57	81.35	81.48	81.47	82.68	81.40	82.44	83.11

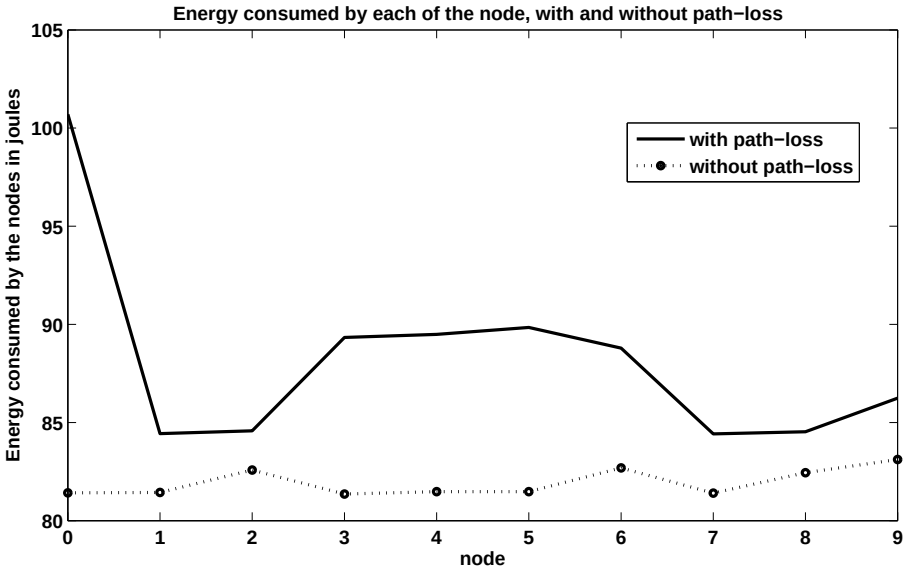

Figure 5.12: Energy consumed by each node with and without path loss

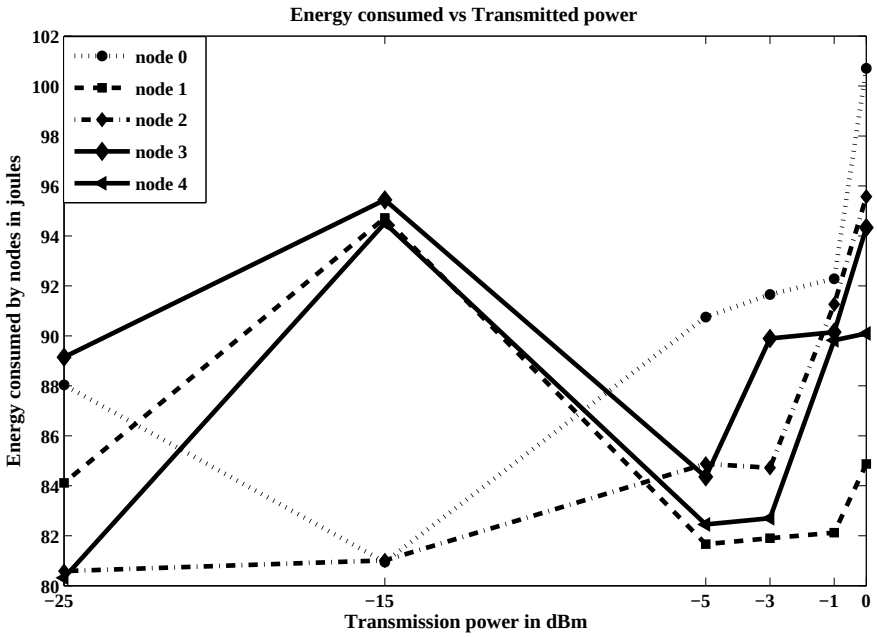

Figure 5.13: Energy consumed vs. transmitted power for nodes 0-4

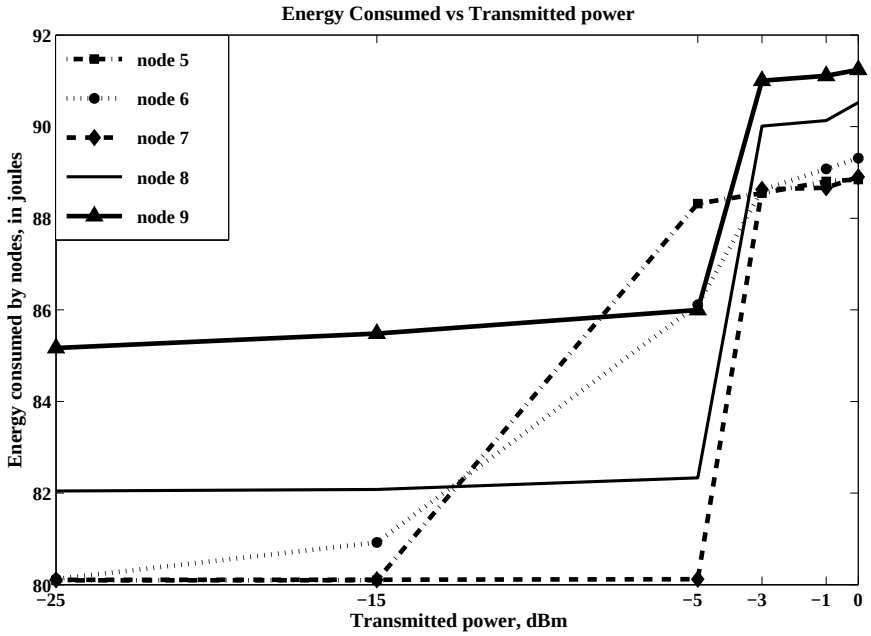

Figure 5.14: Energy consumed vs. transmitted power for nodes 5-9

difference of energy consumed by each node (node 0 to node 9) when path loss is considered for the study and when path loss is ignored.

It is evident that the lifetime of the nodes is heavily dependent on the impact of the path loss, and ignoring the effect of path loss would be an optimistic assumption when energy consumed by each node is considered. This is because, when the effects of path loss are not considered, problems associated, retransmissions and the way this phenomena affects the energy consumption are not taken into account. However these factors affect the life time of the node. Node 3 consumes 13 joules of more energy due to path loss, when compared to no path loss.

Similarly, the impact of transmission power on the energy consumed by each of the nodes is also presented. Figure 5.13 and 5.14 shows the life time of the nodes 0 to 4 and 5 to 9 respectively, considering the impact of path loss for different transmission powers. Transmission power is varied from -25 dBm to 0 dBm, the energy consumption of the nodes is increased as there is increase in the transmission power. For node 7, as the transmission power

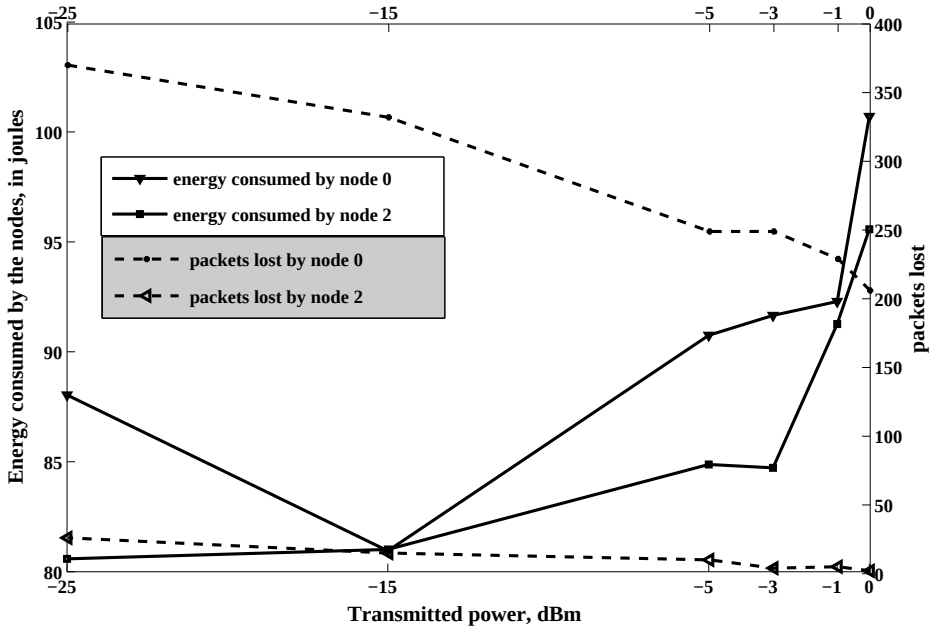

Figure 5.15: Energy consumed vs. transmitted power vs. packets lost

is increased from -25 dBm to 0dBm, the energy consumed by the nodes also increases from 80.1 joules to 88.9 joules.

The transmission rate of the traffic flow is regulated by the transport layer in order to mitigate the congestion in the network. Congestion is closely tied to the local contention in wireless networks. The physical layer controls the contention as well as the channel rate through transmit power control. As a result, equilibrium is reached among nodes in the network. The application layer information, i.e. the contention of the information that is carried by each of the sensor node has direct impact on the design of communication protocols. The trade-off between traditional performance measures such as packet loss and residual energy is presented in Figure 5.15. The dotted lines represent the packets lost and the straight lines represent the energy consumed by each node. As the transmission is decreased from 0 dBm to -25 dBm, there is a gradual increase in amount of packets lost. For node 0, as the transmission power is decreased from 0 dBm to -25 dBm, the number of packets lost increases to 370, from 206 and the energy consumed increases

to 100 joules from 88 joules. Because of the retransmissions, more energy is consumed by the nodes. But the increase in transmission power does not necessarily mean increase in the life time as there are no retransmissions. When the tradeoff between the packet loss and the energy consumed is analysed, it can be seen that the optimum transmission power should be between -15 to -5 dBm where the energy consumption is less than 95 joules and packet loss is less than 200 packets.

5.5.1 Summary and Recommendations

The dynamics of the wireless channel impact the channel quality throughout the lifetime of the network. Although the distance between two nodes does not change (in case of static nodes), the random effects of the wireless channel result in significant fluctuations in communication quality, thereby affecting the lifetime of the network. This drastically affects the quality of the routes that are constructed based on static decisions. The interdependencies of each parameter are analysed in detail in order to provide efficient solution in terms of both performance and cost. Incorporating network layer details along with physical layer information into the MAC layer design improves performance in WSNs. In order to validate the expressivity of the A4WSN modelling languages and to exercise the provided extension points, an analysis plug-in called PlaceLife has been developed, and is presented in this chapter. It is necessary to combine path loss computations used in physical layer, with information from upper layers such as application layer for a more realistic evaluation. In this chapter, a simulation based case study based on home automation system is presented that uses path loss model and application layer information in order to predict the network lifetime. In order to show the usefulness and effectiveness of the approach presented, numerical results along with their analyses is presented. Results presented show that when path loss is introduced, increasing the transmission power is needed to reduce the amount of packets lost. This presents a trade-off between the residual energy and the successful transmission rate when more realistic settings are employed for simulation. It is a challenging task to optimise the transmission

power of WSNs, in presence of path loss, because although increasing the transmission power reduces the residual energy, it also reduces the number of retransmissions required. Power control algorithms encounter a tradeoff in terms of reducing interference and increasing network lifetime vs. reducing network connectivity. By regulating the transmission power of each node, certain network properties such as connectivity, interference, latency, and lifetime can be maintained as necessitated by the application. The cross-layer module concept results in significant gains in performance in terms of energy consumption, throughput, and efficiency in WSNs.

Chapter 6

Clustering Approaches

6.1 Introduction

A WSN consists of a large number of sensor nodes, densely deployed over an unattended area either close to or inside the targets to be observed. These sensor nodes periodically monitor or sense the conditions of the targets, process the data, and transmit the sensed data back to a base station. All of the sensor nodes collaborate together to form a communication network for providing reliable networking service. To provide extensive area coverage, a large number of nodes are required. Moreover, to provide a centralised management system of nodes, clustering algorithms are provided as an effective means to extend lifetime and manage WSN's. Clustering algorithms limit the communication in a local domain and transmit only the necessary information to the rest of the network. In order to support data aggregation, to minimise the total number of messages exchanged between nodes and hence to save energy through efficient network organization, a group of nodes form a cluster and the local interactions between these nodes are controlled with help of a cluster coordinator, known as a cluster head (CH). Figure 6.1 depicts this structure. Clustering provides a structure that can be leveraged to limit the scope of the routing algorithm reaction to changes in the network environment. A number of member nodes which generally communicate the

collected data to the CH, which is aggregated and fused to conserve energy by the CH. Based on this architecture, several hierarchical routing protocols have been developed to address energy efficiency and scalability. Thus, node clustering, which groups nodes into clusters, is critical to facilitate practical deployment and operation of WSNs.

In this chapter, the major issues and challenges in node clustering for WSNs are discussed and a variety of state-of-the-art clustering techniques are introduced and discussed in detail. Although, minimising the energy consumption and extending the lifetime of the network is possible with the introduction of clustering techniques as they decrease the contention through either power control or node scheduling, however, scalability is still an issue. Hence the optimality of the cluster size still needs to be thoroughly investigated. In this chapter, an unequal clustering algorithm for wireless sensor network based on HEED [Younis and Fahmy, 2004b] is proposed. A common problem in equal based cluster in sensor networks is the hot spot problem. The proposed algorithm presented is an attempt to mitigate the hot spot problem. Also, the bottlenecks in the network in terms of cluster size scalability, especially while addressing variety of high packet sending rate and real-time applications, such as wearable heart rate and physical activity monitors and holster monitors is presented. This directly effects the volume of traffic, the CH can handle. Most existing node clustering algorithms for WSNs are tailored from the algorithms originally designed for traditional ad hoc networks. They are primarily focused on scalability and energy conservation in WSNs. A simulation based case study is presented that uses path loss model and application layer details along with the clustering protocols employed to predict the impact of path loss models on the network life time, encouraging realistic WSN lifetime estimations and performance evaluations. In the next section, most popular clustering algorithms and a variety of state-of-the-art techniques are presented.

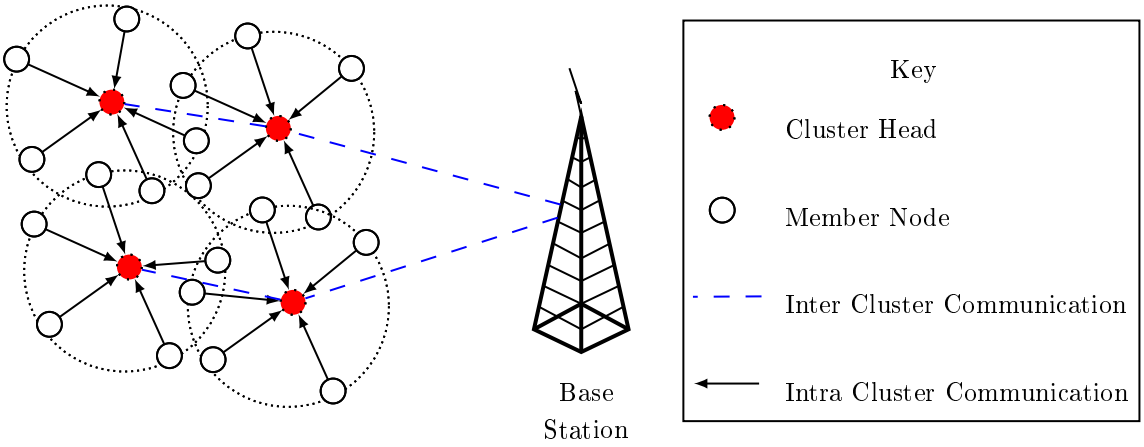

Figure 6.1: Data flow in a clustered network

6.2 Clustering Protocols

HEED

The hybrid energy-efficient distributed clustering (HEED) algorithm combines residual energy of a node and node degree or node proximity to its neighbours to form equal sized clusters. The main goal is to minimize the energy consumed for communication by forming clusters in a distributed fashion. This is performed by combining transmit power control i.e. nodes with high residual energy are selected as CHs. Also, the intra cluster communication cost, which can be cluster density, allows a node to join a CH with the least number of nodes so as to reduce the load of the intra cluster traffic on the CH, is also considered in the cluster formation [Younis and Fahmy, 2004b].

The HEED algorithm is run by each node and is in 3 stages: In the Initialisation stage, an initial percentage of CH among N nodes is set (C_{prob}) which has no impact on the final number of CH to be formed at the end of the algorithm and as such is only necessary to limit the initial number of broadcast. Each node calculates its probability (CH_{prob}) of becoming a CH. The CH_{prob} is not allowed to fall below a certain threshold p_{min} in order for the algorithm to terminate in $O(1)$ iterations. The HEED clustering mech-

anism is performed with a period of $T_{CP} + T_{NO}$ seconds, where, T_{CP} is the time taken for the clustering protocol to converge, and T_{NO} is the network operational interval i.e. time taken for normal network activities.

In the Repeat stage, those nodes that could not join a CH, elect to become a tentative CH and send an announcement. This phase iterates itself and each time the CH_{prob} value doubles until it becomes 1. During the iterations, the node can also decide to find a CH instead of becoming one itself. The clustering algorithm is initiated by CH selection. The probability by which each node determines its probability to become a CH is:

$$CH_{prob} = max\left(C_{prob}\frac{E_{residual}}{E_{max}}, P_{min}\right) \tag{6.1}$$

where, $E_{residual}$ and E_{max} are respectively the residual and maximum energy of the node. CH_{prob} depends on the relative lifetime of a node, bounded by a threshold p_{min}, to ensure convergence within a limited number of iterations.

In the Final stage, a node decides its status to become a final CH for the current round or joins the least cost cluster. Once the clustering process is over, the network enters a data transfer phase. Clustering will occur again after some time in order to rotate the role of the CH and thus balance the energy levels in the network. In this phase, each node of a cluster forwards data to the CH which in turn forwards the aggregated data of its members in a multi-hop fashion (CH to CH) till the base station (BS) is reached.

The problem here is that those nodes nearer to the BS deplete their energy faster than those located further away, causing hotspots. This excessive inter-cluster traffic near the BS causes the nearby nodes to die earlier reducing the overall network lifetime.

EEUC

Energy Efficient Unequal Clustering (EEUC) is a distributed competitive algorithm, proposed in order to balance the energy consumption among clusters by wisely organizing the network via clustering and multi-hop routing,

as there exists a hot-spot problem i.e., the CHs closer to the BS are burdened with heavy relay traffic and tend to die early, leaving areas of the network uncovered and hence leading to network partition [Li et al., 2005a]. To address such an issue, EEUC mechanism for periodical data gathering is proposed. In EEUC, the cluster sizes near the sink node are much smaller compared to the clusters far away from the sink, in order to save energy in intra-cluster communications and inter-cluster communications. Thus, CHs closer to the BS can preserve energy from the inter-cluster data forwarding.

EEUC is a distributed cluster heads competitive algorithm, which uses the competition radius formula given below, in order to create unequal clusters. Since the lifetime of the leaders closer to the BS is more critical, the clusters further away have larger sizes compared to the clusters close to the BS.

$$R_{comp} = \left(1 - c\left(\frac{d_{max} - d(s_i, BS)}{d_{max} - d_{min}}\right)\right) R^0_{comp} \tag{6.2}$$

R^o_{comp} is the maximum competition radius which is predefined. In this work it is defined as the diagonal distance of the sensing grid area divided by 10. d_{max} and d_{min} are the maximum and minimum distance between sensor nodes and the base station; c is a constant coefficient between 0 and 1.

These phases are repeated during the lifetime of the network and are called rounds. A round starts by triggering the clustering mechanism and after clusters have been formed, the network goes into a data exchange phase. It ends when all cluster heads have transmitted the aggregated data of their members to the base station once. Though the nodes join clusters of unequal sizes, eliminating hotspot problems, EEUC may produce lone nodes as the CH election is probabilistic [Gong et al., 2008].

6.2.1 LEACH and Unequal LEACH

Through a cluster-based operation, the LEACH protocol aims to minimise energy consumption in WSNs [Heinzelman et al., 2000]. The goal of LEACH is to dynamically select sensor nodes as cluster heads and form clusters in

the network. Aggregation os performed at the cluster head, to which the communication inside the clusters are directed to. The cluster head then directly communicate with the sink to forward all the collected information each cluster. LEACH also changes the cluster head role dynamically such that the high-energy consumption in communicating with the sink is spread to all sensor nodes in the network.

The operation of LEACH is controlled through rounds, consisting of several phases. In each round, each cluster formation stays the same, and at the beginning of each round the cluster heads are selected. A round is separated into two phases, the set up phase and steady state phase. During the set up phase, cluster heads are selected, clusters are formed, and the cluster communication schedule is determined. During the steady state phase, data communication between the cluster members and the cluster head is performed. The duration of the steady state phase is longer than the duration of the set up phase in order to minimize the overhead. There are three phases in set up phase of LEACH: advertisement, cluster set up, and schedule creation. The cluster head selection (randomly select sensors as cluster heads during the beginning of each round) is performed through the advertisement phase, where the sensor nodes broadcast a cluster head advertisement message. Once the sensor nodes receive the advertisement message from other nodes, they determine the cluster to which they belong to. If a node receives an advertisement from a single cluster head, then it automatically becomes a member of that cluster. However, if a sensor node receives advertisements from multiple cluster heads, the cluster selection is performed based on the signal strength of the advertisement from the cluster heads to the sensor nodes. Once the cluster formation is completed in the set up phase, LEACH switches to the steady state phase. In the steady phase, the sensor nodes can begin sensing and transmitting data to the cluster heads. The cluster heads also aggregate data from the nodes in their cluster before sending these data to the sink. At the end of the steady state phase, the network goes into the set up phase again to enter into another round of selecting the cluster heads. As a result, energy consumption due to the cluster head duty

is equally distributed among sensor nodes. The cluster-based operation of LEACH improves the energy efficiency of WSNs. During the steady state phase, only the cluster heads are active all the time. A cluster member in a cluster is active only during its allocated time slot and the set up phase. Consequently, the energy consumption of a regular node is minimized significantly. Since LEACH performs periodic cluster head selection, the energy consumption burden of the cluster head nodes is also shared.

An LEACH variant scheme, unequal LEACH or improved LEACH is proposed [Ren et al., 2010a] in order to solve the hot spot problem by creating unequal sized clusters by varying the competition radius. Smaller clusters will be formed near the base station while larger ones will be created as they are further away from it.

6.2.2 UHEED

In multi-hop clustering, nodes nearest to the BS tend to deplete their energy the fastest since they are burdened with heavy relay traffic from the rest of the network in addition to their own intra-cluster traffic share. Those nodes closer to the BS tend to die earlier than the rest and as a result, sensing coverage gets reduced and network partitioning becomes apparent, [Kim et al., 2008; Li et al., 2005b; Xuhui et al., 2009; Zhao and Wang, 2010b] which is defined as the hot spot problem. Nevertheless multi-hop data transmission from source to BS is usually more energy efficient due to the nature of the wireless channel [Zhao and Wang, 2010a].

In this chapter, an unequal clustering algorithm (UHEED), based on the HEED algorithm [Younis and Fahmy, 2004b], is proposed. UHEED creates unequal sized clusters based on the distance of the CH from the BS. The further away a cluster head is from the BS, the larger will be its competition radius and hence the cluster size will be bigger compared to those clusters formed nearer to the BS. By creating unequal sized clusters, the amount of intra-cluster traffic is considerably reduced for the CH's nearer to the BS. UHEED makes use of competition radius equation 6.5 to create unequal

clusters and 6.1 to calculate the probability by which each node determines its probability to become a CH. While in HEED, each CH employs the same competition radius, irrespective of its distance from the BS, therefore on an average have the same number of nodes in a cluster. UHEED on the other hand, uses competition radius equation 6.5 to create smaller clusters closer to the BS.

6.2.3 Network Model

The network model introduced in this chapter uses a two dimensional representation of the environment and the nodes are deployed randomly following a uniform distribution. The following assumptions are considered : (i) all nodes are homogeneous in terms of energy, communication and processing capabilities;(ii) each node is identified with a unique ID; (iii) nodes can transmit at various power levels depending on the distance of the receivers; (iv) nodes are not mobile that is they remain stationary after the uniformly distributed deployment process; (v) communicating nodes can establish the distance among them[1]; (vi) all nodes know their distance from the base station.

The BS is located away from the sensing grid with no energy concerns at all, and it is considered to be a node with enhanced communication and computation capabilities. The BS is not mobile. The data captured in a cluster is highly correlated, therefore it can be aggregated before being transmitted to the base station.

A network operation model similar to that of [Younis and Fahmy, 2004b] consisting of multiple rounds is considered. A round starts by triggering the clustering mechanism and after clusters have been formed, the network goes into a data exchange phase. This includes intra-cluster communication where each sensor node sends exactly one message to its cluster head and inter-cluster communication where each aggregated data is sent by the cluster head

[1]Usually nodes estimate the approximate distance by the strength of the signal received, since the transmission power level is known (unless there is multi-path fading problem)

to the BS (multi-hop data transmission among cluster heads is performed). The round ends when all aggregated data sent by the cluster heads are received at the base station.

The radio model employed uses both the free space and the multi-path channel model and assumes error-free communication links. The simulation parameters used are similar to those in [Younis and Fahmy, 2004b], in order to compare the results with the existing literature. A sensor spends $E_{elec} = 50nJ/bit$ [Younis and Fahmy, 2004b] to run the transmitter or receiver circuitry. The energy spent by the transmitter amplifier E_a will depend on the distance d between the sender and the receiver: $E_a = E_{fs}$ assuming a free space model when $d < d_0$ and $E_a = E_{mf}$ assuming a multipath model when $d \geq d_0$, where $d_0 = 75m$ is a constant distance. $E_{fs} = 10pJ/bit/m^2$ and $E_{mf} = 0.0013pJ/bit/m^4$. In order to transmit a k-size packet over a distance of d using the above radio model, the amount of energy consumed for transmission E_{T_x}, can be calculated as:

$$E_{T_x} = (E_{elec} \times k) + (E_a \times k \times d^n), \tag{6.3}$$

where, $n = 2$ for the free space model and $n = 4$ for the multipath model. The amount of energy E_{R_x} spent to receive a k-bit size message is:

$$E_{R_x} = (E_{elec} \times k) \tag{6.4}$$

6.2.4 Simulation Model

UHEED creates unequal sized clusters based on the distance of the CH from the BS. Since the lifetime of the CHs closer to the BS is more critical, the clusters further away have larger sizes compared to the clusters close to the BS.

$$R_{comp} = \left(1 - c\left(\frac{d_{max} - d(s_i, BS)}{d_{max} - d_{min}}\right)\right) R^0_{comp} \tag{6.5}$$

R^o_{comp} is the maximum competition radius which is predefined. In this work

it is defined as the diagonal distance of the sensing grid area divided by 10. d_{max} and d_{min} are the maximum and minimum distance between sensor nodes and the base station; c is a constant coefficient between 0 and 1.

These phases are repeated during the lifetime of the network and are called rounds. A round starts by triggering the clustering mechanism and after clusters have been formed, the network goes into a data exchange phase. It ends when all cluster heads have transmitted the aggregated data of their members to the base station once. The simulation performed in this study consists of 2 phases namely clustering and data exchange. These are performed until the network is dead. The network is considered dead when all nodes have depleted 99.9% of their energy. Network lifetime is based on rounds rather than clock time, and the simulation model used is event triggered. In other words, the simulation clock is always set to the time of the next event until the network dies.

The simulation program is first validated by using the numerical results presented in the existing literature [Younis and Fahmy, 2004b]. For the validation, a grid with dimensions 2000 × 2000 metres is considered and 1000 nodes are deployed. The cluster radius is taken from $20m$ to $400m$ and each experiment value is obtained for an average of 1000 runs. The numerical results shows that the implementation of the HEED algorithm exhibit similar behaviour as in[Younis and Fahmy, 2004b].

6.2.4.1 Equal sized clusters: Existence of hot spots

The results presented in this section are related to the HEED equal clustering algorithm. More specifically, the HEED algorithm has been run for one round, and figures 6.2 and 6.3 clearly show that cluster heads nearer to the base station have lower residual energy compared to that of cluster heads further away. The results presented are for cluster radius's of 20m and 50m, however the behaviour is the same for different cluster sizes as well. From Figure 6.3, as the distance from the base station increases, the residual energy of the nodes increases. More precisely, after one round, the residual energy

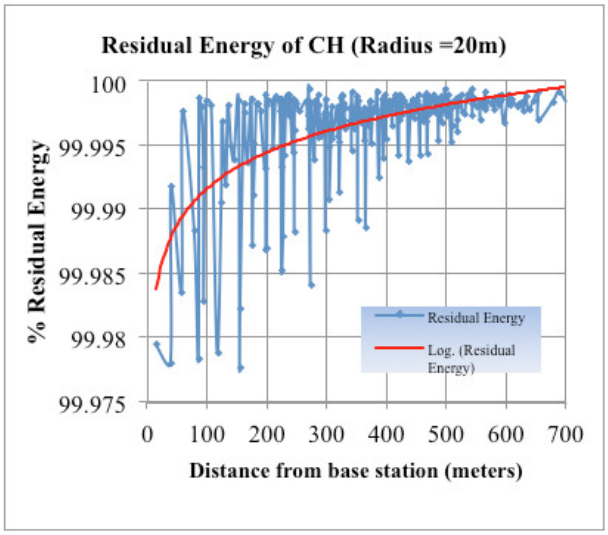

Figure 6.2: Residual energy of cluster heads (r=20m)

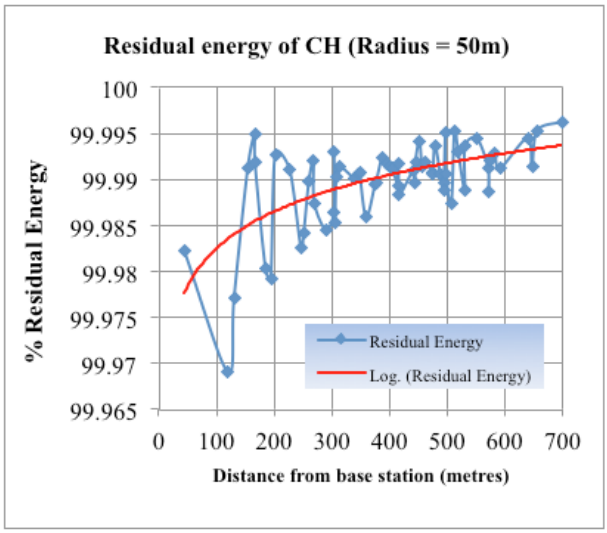

Figure 6.3: Residual energy of cluster heads (r=50m)

is 99.98% for the nodes 100m away from the BS, compared to 99.9925% for the nodes 700m away.

6.2.5 Network Lifetime

The network lifetime for UHEED protocol is evaluated by running simulations with various parameters of grid size and number of nodes.

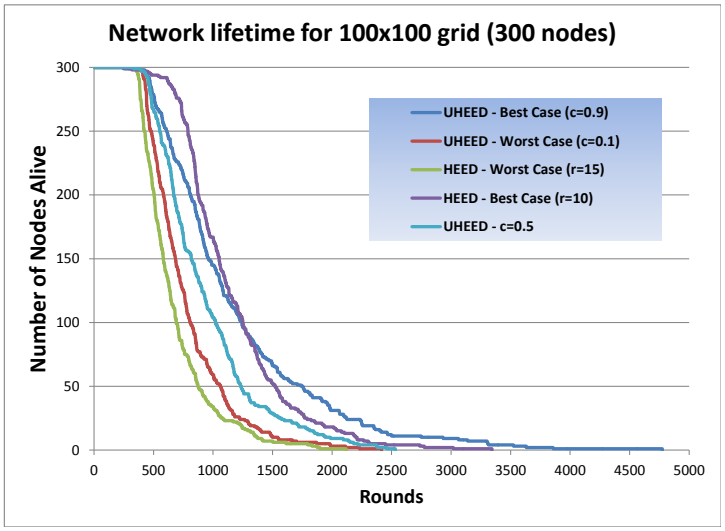

Figure 6.4: Network lifetime for 100x100 grid with 300 nodes

To have a fair comparison for UHEED and HEED, the same parameters from [Younis and Fahmy, 2004b] have been used. The base station is located at lower right side of the grid, $E_{elec} = 50nJ/bit$, E_a =$10pJ/bit/m^2$, number of nodes is varied from 300 upto 1000 and the initial Energy is 2 Joules. More specifically, The following two settings are used: (A) a grid size of 500m x 500m with 1000 nodes; (B) a grid size of 100m x 100m with 300 nodes. Same number of cluster heads in both UHEED and HEED have been used. This ensures that the two algorithms perform the same number of hops in the inter-cluster communication. Figures 6.7 and 6.4 show the best and worst case scenario, for network life time, for UHEED and HEED for:

Case(A): The results show that UHEED outperforms HEED in the best scenario with $c = 0.8$ for UHEED and $r = 35m$ for HEED; but for the worst case, UHEED and HEED both follow a similar pattern with the last node dying after around 500 rounds. In the best case scenario, the last node for UHEED dies after round 3325 and for HEED, it dies after 900 rounds which is evident in Figure 6.7. Overall, there is a 250% increase in network lifetime for UHEED in the best case scenario.

Case(B): UHEED outperforms HEED in both the worst and best scenarios. In the best case scenario, the last node in UHEED dies after 4750 rounds, where as, in HEED, it dies after 3340 rounds, which can be observed in

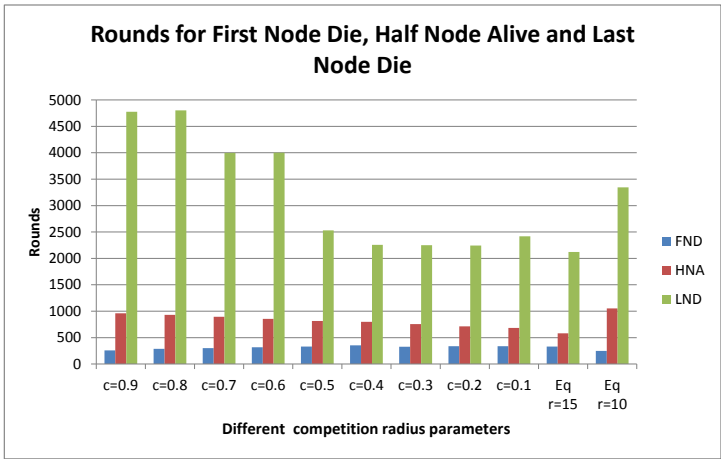

Figure 6.5: Node lifetime analysis for 100x100 grid with 300 nodes

Figure 6.4. Overall, there is an increase of more than 40% network lifetime for UHEED in the best case scenario.

Figures 6.8 and 6.5 illustrates the residual energy of the network when the first node die and when half of the network is still alive for the for the cases (A) and (B) respectively. It is worth mentioning that in these set of figures each set of clustered column labelled with parameter c is related with UHEED, and similarly each clustered column using parameter r is related with HEED. Please note that when network lifetime is considered for UHEED algorithm, systems with higher c values outperforms the systems with lower c values. The main reason of that is when the cluster is designed to balance the residual energy, the energy consumption is also balanced in a way to make sure each cluster head has similar lifetimes. When one of the head nodes lives longer on the other hand, this would increase the network life time since the network is only considered dead only when all nodes have depleted 99.9% of their energy. Figures 6.9 and 6.6, illustrates the residual energy of the network when the first node die and when half of the network is still alive for the above scenarios.

To find optimum parameters and protocols, UHEED, LEACH and unequal LEACH are compared in terms of network lifetime, and residual energy levels for different competition radius parameters, the same parameters from [Heinzelman et al., 2000] and [Ren et al., 2010b] have been used. More specifically, the

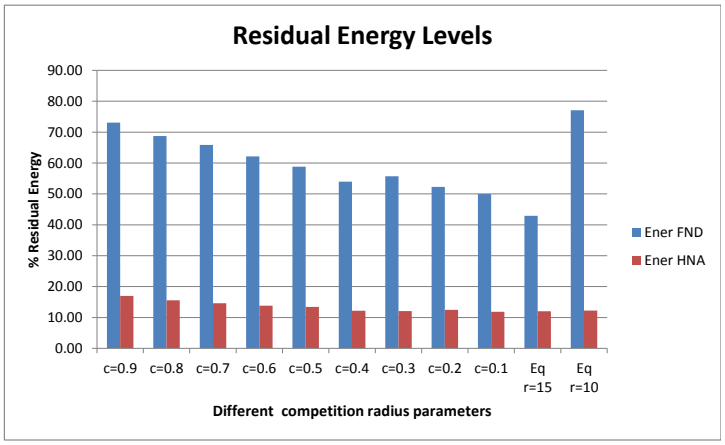

Figure 6.6: Node residual energy levels for 100x100 grid with 300 nodes

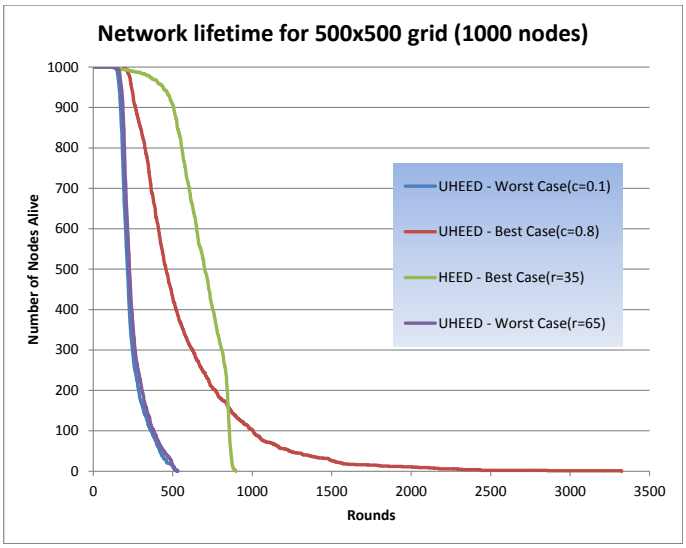

Figure 6.7: Network lifetime for 500x500 grid with 1000 nodes

base station is placed outside the grid, $E_{elec} = 50nJ/bit$, E_a =$10pJ/bit/m^2$, number of nodes 400, with an initial energy of 0.3Joules.

UHEED is compared to unequal LEACH [Ren et al., 2010b] and results are presented in Figure 6.10. The best and worst case for UHEED and unequal LEACH are considered. For unequal LEACH, to have a fair comparision, the parameters from [Ren et al., 2010b] are used, with a grid size of 200 by 200 and 400 nodes. In order to compare UHEED and unequal LEACH, the data exchange phase of UHEED has been modified to a single-hop data transmission since unequal LEACH is a single-hop protocol.

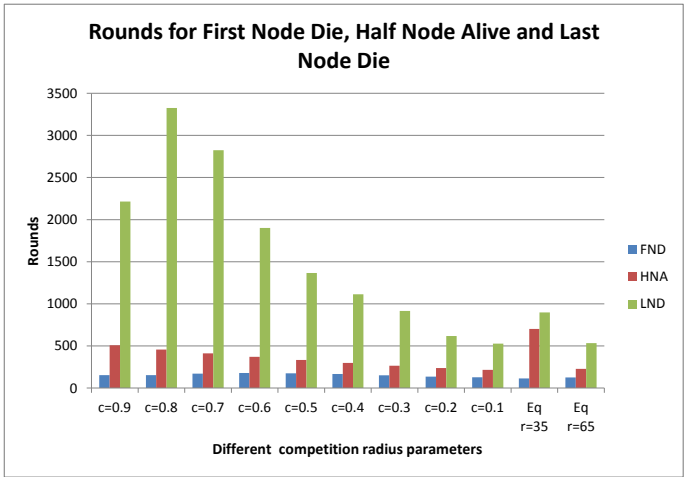

Figure 6.8: Node lifetime analysis for 500x500 grid with 1000 nodes

Figure 6.11 illustrates the round at which the first node dies, half of the nodes are alive and last node dies for unequal LEACH, original Leach and the different values of c from 0.1-0.9 for UHEED. In the simulation study between UHEED and unequal LEACH, it can be observed from Figure 6.10, UHEED outperforms unequal LEACH by a factor of more than 100% when network lifetime is considered.

Figure 6.12 shows the residual energy for UHEED, unequal LEACH and LEACH with respect to First Node Dead (FND) and Half Node Alive (HNA). The residual energy is obtained by calculating the residual energy of the entire network. As can be seen from the Figure 6.12, in UHEED after the first node is dead, the overall residual energy level for all the cases from $c = 0.1$ to $c = 0.9$ is much higher than LEACH or unequal LEACH. Also, it is observed that when half of the nodes are alive, the residual energy level in case of UHEED is comparatively higher than LEACH and unequal LEACH. Hence, from the results seen in Figure 6.12 for residual energy levels and Figure 6.10 for the network lifetime, it is seen that the lifetime degradation of UHEED is graceful. This means that not many nodes die very quickly and then the network has very few nodes which are alive for a longer duration, but, as observed in Figure 6.12, after half the number of nodes are dead in the case of UHEED, there is still higher residual energy level available for the rest of the nodes to continue operation with respect to LEACH and unequal

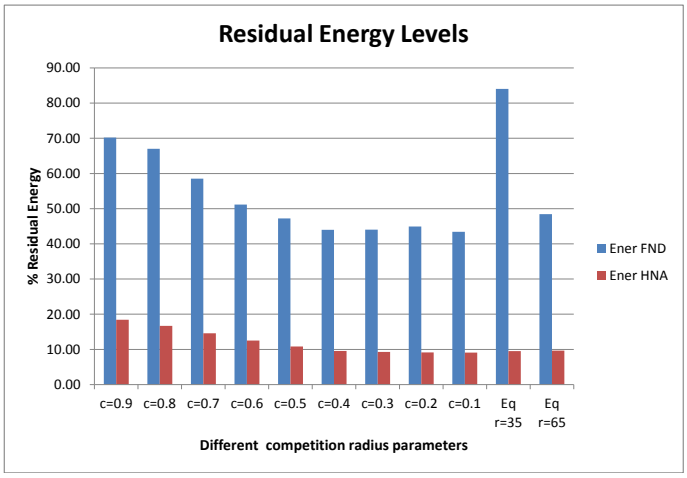

Figure 6.9: Node residual energy levels for 500x500 grid with 1000 nodes

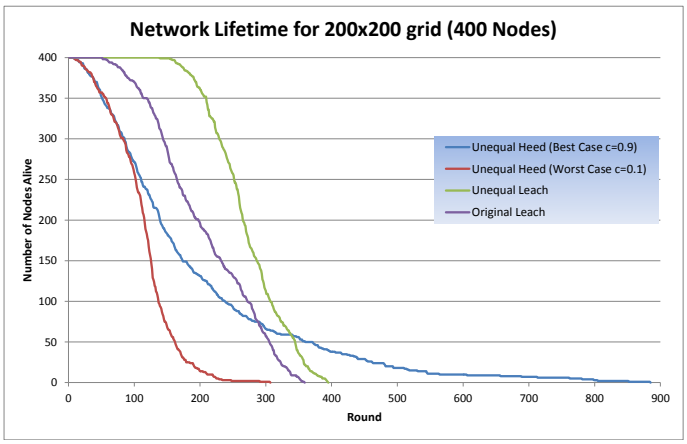

Figure 6.10: Network lifetime comparison for HEED, UHEED, LEACH and unequal LEACH

LEACH.

6.3 Optimal Cluster Size

In this section, the system description along with the assumptions considered, as well as the queue model of the system is presented. A cluster network with one CH coordinating the cluster operations is considered. Sensed information at the nodes is forwarded to the CH which finalises cluster aggregation and transmit all the information to the sink either directly or through inter-

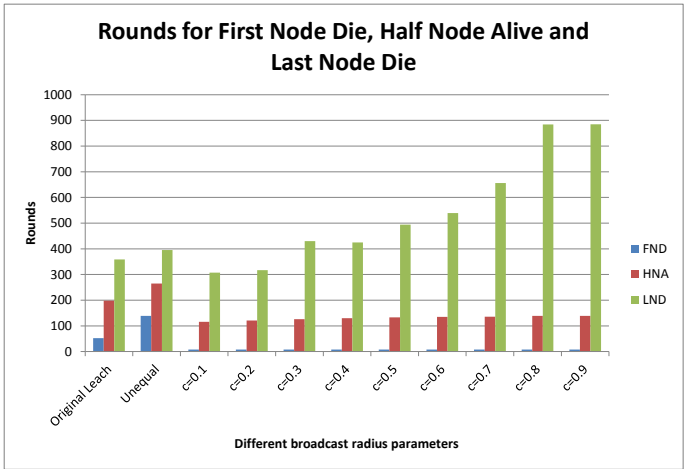

Figure 6.11: Node lifetime analysis for LEACH and UHEED

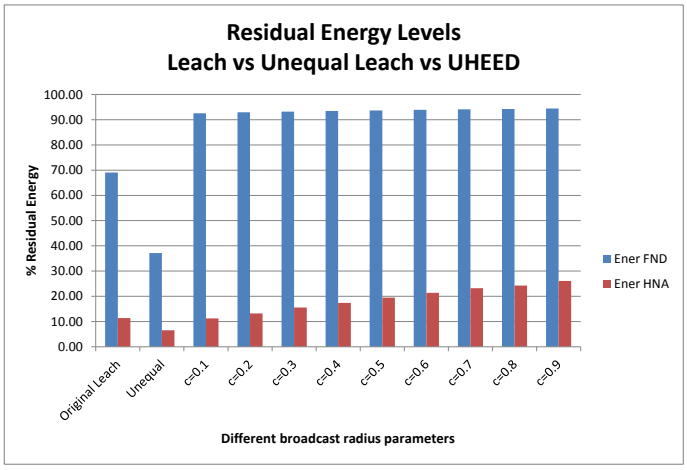

Figure 6.12: Node residual energy levels for LEACH and UHEED

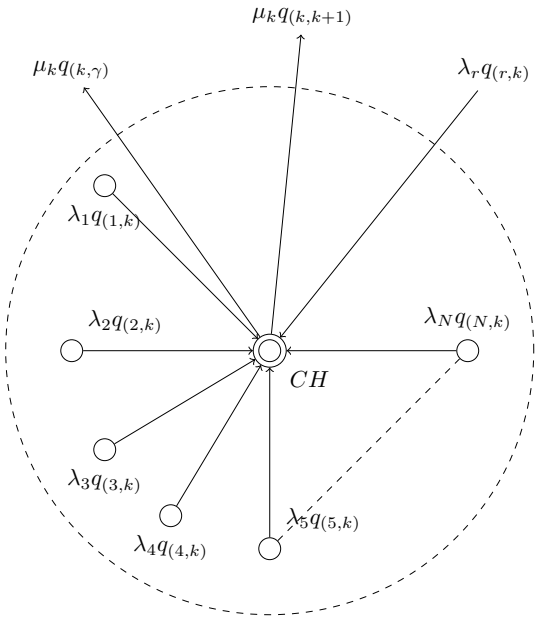

Figure 6.13: Queuing model of a single CH

mediary CHs. We assume that all the sensor nodes are connected directly to the CH, hence they communicate via their CH. It is assumed that at least one path always exists towards the sink [Chiasserini and Garetto, 2004].

The resulting job arrivals (packets sent to the base station) at the CH is the collection of jobs from random nodes, (λ_1, λ_2... λ_N) in the cluster. The behaviour of each CH is considered as an open queue network using M/M/1 queues [Omondi et al., 2013a]. A single CH's behaviour can be analysed with the queuing model shown in Figure 6.13. The internal arrivals to the CH ($k = 0$) are arrivals within the cluster (from nodes 0,1,2...N). The mean arrival rate (λ_k) at the CH can therefore be given by $\lambda_k = \sum_{n=0}^{N} \lambda_n$, where n = 0,1,....N. It is assumed that the resulting superposition of all the job arrivals at node k follows Poisson distribution with mean arrival rate λ_k packets/sec. The analysis performed in the next section shows that the assumption holds.

The external arrivals are from the other CHs, forwarding their data to the sink, through node k. Once the jobs are processed at node k, they are transmitted directly or forwarded to the sink through an intermediary node r (node r represents the next CH towards the sink). Node 0 ($k = 0$) represents

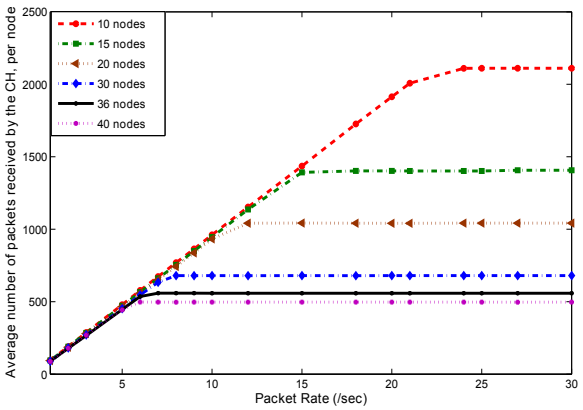

Figure 6.14: Comparison of packet arrivals at CH vs Packet rate of cluster nodes

the CH. At node r the process is similar to that at node k. The external arrival to the CH (from other CHs) are arrivals with rate λ_r. The other arrivals are originated from the sensor nodes forwarding their data to CH. The rate of traffic from node r within the cluster to node k is $\lambda q_{(r,k)}$. The operation is assumed to be similar at all other CHs. Packets are handled on first come first served (FCFS) basis. In the following section, the validity of the model is analysed using Castalia simulator and results are presented.

Numerical results are presented for the model considered. The following parameters are used throughout this section, unless otherwise stated. A CC2420 chip, compatible with 802.15.4, is used to provide wireless communication, operating at 2.4 GHz and providing a data rate of 250 kbps. The packet size is considered to be 105 bytes [LatrÃľ et al., 2005]. TMAC, which provides both collision avoidance and reliable transmission is used as the MAC protocol. Each simulation lasts for 100 sec to reach steady state and the results are taken over an average of 100 runs.

In order to analyse the effects of packet rate and number of nodes on the cluster, simulation studies are performed by varying the number of nodes and packet rates in each run. The number of nodes in the cluster are varied from 10 to 40. All the nodes are sending the sensed information to the CH at a constant rate which is varied from 1 packet/sec to 30 packets/sec. The results presented are in good agreement with the two different analytical solution

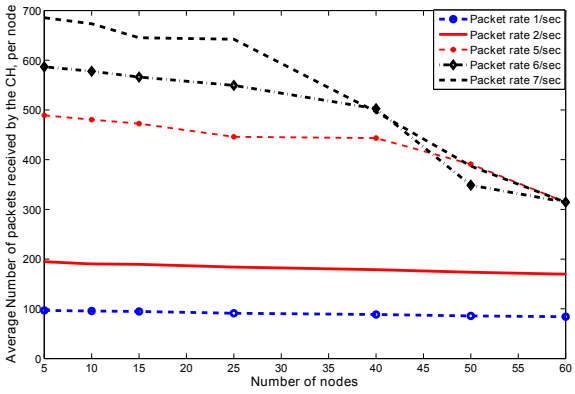

Figure 6.15: Comparison of packet arrivals at CH vs number of cluster nodes

approaches for performability modelling of a WSN cluster, presented by the authors in [Omondi et al., 2013a]. Figure 6.14 shows the average number of packets received by the CH, from each node as a function of varying nodes and packet rates. The results presented show that higher packet rates can be accommodated by fewer nodes to attain saturation (maximum utilization reached), as compared to the network with low data rates and higher number of nodes. In other words, to scale out the network for wider coverage a trade-off between number of nodes is appropriately considered against the aggregate packets received at the CH.

The saturation points for possible number of cluster nodes for packet rates chosen (Figure 6.15), are in agreement with the work presented by the authors in [Omondi et al., 2013a]. The number of nodes has been varied from 5 to 60, and the packet rates are varied from 1 packet/sec to 7 packets/sec. At low packet rates (say 1 packet/sec, 2 packets/sec), saturation levels are reached much later, with more number of nodes in the system. At higher packet rates, saturation levels are reached earlier with less number of nodes. Hence, this analysis can be used to specify the size of a cluster, when specific flow (data rates) is expected from different types of applications. In other words, it is a key issue that affects the practical deployment of clustering techniques in sensor network applications.

6.4 Affect of path loss on Clustering protocols

The limited energy resources of sensor nodes are one of the most important constraints. Hence, a more realistic approach for analysing WSNs and evaluating its performance in terms of energy efficiency would be required for designing and deploying WSNs, under realistic conditions.

Consequently, the Medium Access Control (MAC) layer is crucial as it dictates the state of the radio and hence, the power consumption of the node is based on the time the radio is on (either listening, transmitting, or receiving), and how long the radio stays in each of the states. Most of the existing research using simulators do not consider energy consumed for listening for the performance evaluation in terms of lifetime, leading to optimistic performance evaluation [Chang and Kuo; et.al.; Ever et al., 2012; Heinzelman et al., 2002; Heinzelman et al.; Lindsey and Raghavendra; Manjeshwar and Agrawal, 2001; Younis and Fahmy, 2004a; Zahmati et al.]. Wireless channel is another dominant factor in the performance of communication protocols in WSNs. The low power communication capabilities of sensors and the rather limited capabilities of low-cost transceivers result for a significant impact on the higher layers. Hence, the effects of the wireless channel cannot be only confined to the physical layer.

Path loss is the attenuation in power density of an electromagnetic wave as it propagates. Path loss is consequence of many effects such as free-space loss, refraction, diffraction, reflection, aperture-medium coupling loss, and absorption. Path loss is also affected by other factors such as propagation medium (dry or moist air), the distance between the transmitter and the receiver, and the frequency of the signal. When the effects of path loss are not considered, the evaluation of underlying structure can become optimistic, since the problems associated, retransmissions and the way this phenomena affects the energy consumption are not taken into account.

In order to show the usefulness and effectiveness of a realistic approach for analysing WSNs and evaluating the performance in terms of energy efficiency for designing and deploying WSNs, under realistic conditions, the numerical

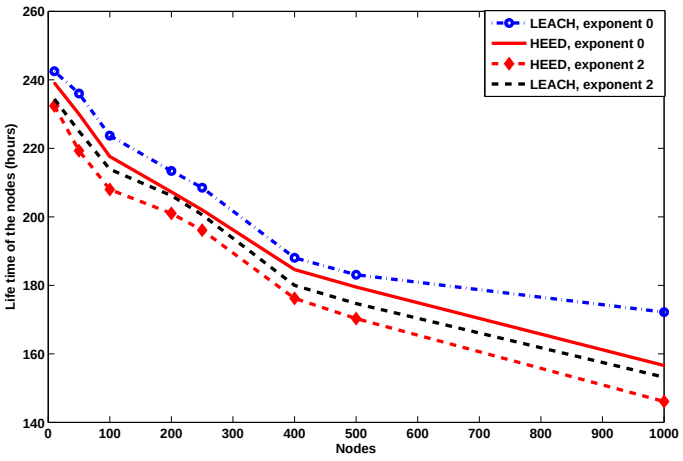

Figure 6.16: Lifetime: LEACH and HEED, with and without path loss

results in Figure 6.16 are presented. LEACH and HEED protocols are implemented in castalia, under realistic conditions. A CC2420 chip, compatible with 802.15.4, is used to provide wireless communication, operating at 2.4 GHz and providing a data rate of 250 kbps, the network size is 50m*50m, and 2 AA battery are used as the energy source for each node. Figure 6.16 is presented for the scenario where the path loss is caused by the obstacles (exponent is two) and compared with the scenario when the path loss is avoided (exponent is zero). The results clearly show that avoiding path loss would cause over estimation of WSN lifetime. More precisely, the life time of a node is 208 hours for LEACH considering the path loss, compared to 219 hours, ignoring the effects of path loss, for 100 nodes. Similarly, for the case of 1000 nodes, the life time of the node is 146 hours, considering effects of path loss, compared to 172 hours, ignoring the effects of path loss for HEED.

6.5 Summary

Node clustering is very important in WSNs because it provides a topology control approach to reduce transmission overheads and exploit data aggregation among a large number of sensor nodes. One critical step in node clustering is to select a set of cluster heads and group the remaining sensor

nodes into clusters with these cluster heads. Most existing node clustering algorithms for WSNs are tailored from the algorithms originally designed for traditional ad-hoc networks. They are primarily focused on scalability and energy conservation in WSNs. Clustering in WSNs has several unique challenges in its deployment, for example, ensuring connectivity among sensor nodes, determining optimal cluster sizes, and dynamically optimizing clustering structures based on the status of cluster members. To achieve optimal performance, it is important to consider the interactions between different layers of the protocol stack. The physical location of each node in a WSN is crucial in terms of both communication and sensing. Connectivity is an important factor considered for sensor deployment as sensor nodes are equipped with limited range radios. Also, as the physical phenomenon with spatio-temporal characteristics play an important role in WSNs, the deployment of nodes also affects the performance of the network.

In this chapter, an unequal clustering algorithm for wireless sensor network based on HEED [Younis and Fahmy, 2004b] is proposed. A common problem in equal based cluster in sensor networks is the hot spot problem. The proposed approach was first to implement the HEED algorithm, proving the existence of hot spot problem, and finally presents an attempt to mitigate it by proposing UHEED. This algorithm uses a competition radius formula which creates unequal clusters as they are further away from the base station. This effectively allows more inter-cluster or relay traffic and less intra-cluster communication for nodes nearer to the base station, hence preventing early death of CHs closer to the BS. Simulation performed on the UHEED algorithm demonstrated the lifetime of the network was increased in all test scenarios compared to HEED, LEACH and unequal LEACH. An interesting study was also conducted regarding the value of the constant c in the competition radius formula. Simulation results showed that for the value of $c = 0.8$, achieved up to almost 250% improvement in the network lifetime when compared to HEED and almost 100% improvement, as compared to unequal LEACH.

Though clustering techniques extend the lifetime of the network, scalability

is still an issue, hence the optimality of the cluster size is thoroughly investigated. In order to have a wider coverage, a trade-off exists between the number of nodes in a cluster being considered and the aggregate packets received at the CH. The results presented show that this analysis can be used to specify the size of a cluster when a specified flow of data is expected from the sensing nodes. In other words, higher packet rates can be accommodated by fewer nodes to attain saturation, as compared to the network with low data rates and higher number of nodes. This directly affects the volume of traffic, the CH can handle. This has been thoroughly investigated in this chapter.

A simulation based case study is also presented that uses path loss model and application layer information along with the clustering protocol employed to predict the network life time. The results presented in Figure 6.16 clearly show that avoiding path loss would cause over estimation of WSN lifetime.

Chapter 7

Intrusion Detection Systems

7.1 Introduction

The remote and unattended operation of sensor nodes increases their exposure to malicious intrusions and attacks. Further, wireless communications make it easy for an adversary to eavesdrop on sensor transmissions. Ensuring the security of both the collected data and the process of data collection is vital for the success of WSNs. Because of the constraints of the particular applications and the resource limitations, the security of WSNs is vastly different from that of conventional wired networks. Research has been conducted in Intrusion Detection Systems (IDS) for wired networks for over two decades, however, wireless area networks and personal area networks have been the focus of research in recent days, as they represent and pose new risks and security challenges. A WSN consists of spatially distributed autonomous sensors that monitor environmental conditions in order to accomplish a task. In applications such as battlefield surveillance and assessment, target tracking, monitoring civil infrastructure such as bridges and tunnels, and assessment of disaster zones to guide emergency response activities, any breach of security, compromise of information, or disruption of correct application behaviour can have very serious consequences. In all these applications, security is one of the main concerns since lives and livelihoods are depending

on the WSN. WSN are vulnerable to different types of security threats that can degrade the performance of the whole network. The main challenges in designing security protocols for WSNs are limited energy source, limited storage of each wireless sensor and limited communication power of the sensor nodes [Dargie and Poellabauer, 2010].

Though sensor networks share some commonalities with a typical computer network, the unique requirements of its own make them a special type of network. Wireless sensor networks pose new security challenges because of their unattended nature and limited resources. Intrusion detection in WSNs present a number of new risks and significant challenges that are not faced by either wired, IEEE 802.11 based networks, or even by mobile ad-hoc networks. Although previous research techniques used in these networks can serve as stepping stones for development of IDS mechanisms for WSNs, due to the resource constrained nature, are not applicable [López and Zhou, 2008]. Although prevention based security measures such as encryption [Jia et al., 2008], authentication, key management and firewalls [Ma et al., 2006], have been used to protect the network from insider attacks that may extract sensitive information, the attacker can physically access the WSN and tamper with the nodes in order to subvert the correct WSN behaviour.

Intrusion Detection Systems can be used to mitigate the problem. They are a second line of defence that analyses the observable behaviour of a system in order to recognise malicious behaviour. IDS analyses the network by collecting sufficient amount of data and detects abnormal behaviour of sensor nodes [Roman, 2006]. Once the attack is detected, the IDSs raise an alarm to inform the controller to take action

There are two main types of intrusion detection techniques: *misuse* and *anomaly*. Misuse detection systems [T.Eckmann et al., 2002] are explicitly programmed to recognise well-known attacks. These systems recognise intrusions by matching the pattern of observed data with the set of predefined (intrusion) signatures. They can perform focused analysis thus having a low false alarm rate. However, they cannot detect unknown types of attacks. Anomaly detection systems assume that an attack will cause deviation from

normal behaviour, thus detection can be done by comparing actual activities with known correct behaviours. Different approaches have been used to model normal behaviour: statistics-based [Javitz and Valdes, 1994], rule-based [Vaccaro and Liepins, 1989] and formal specification [Stillerman et al., 1999]. The advantage of this kind of systems is the ability of detecting unknown attacks. However, it is not easy to define what is a normal behaviour and set up anomaly thresholds in order to have a good detection efficiency and a low positive rate.

Intrusion detection in WSN is a particularly challenging task because of the limited resources of the nodes. Many intrusion detection schemes have been introduced for WSN in the literature. WSNs can operate in two different modes called as continuous periodic sensing and transmission or event-triggered sensing. The decision on which mode of operation to use is highly dependant on the application. For WSNs, while the IDS enhances security, it can shorten the lifetime of the WSN since the IDS may require to run in promiscuous mode [Chen et al., 2007; Filipovic and Datta, 2004]. More precisely in promiscuous mode, each IDS can continuously eavesdrop the radio in order to check the correct behaviour of all other nodes. This solution not only makes impossible to optimise the duty cycle (nodes can never sleep) but also requires the nodes to be in the same range. However promiscuous mode is not really suitable for applications with event-triggered sensing.

Although, there have been some recent developments in the area of IDS for wireless adhoc networks, there is no work reported in the literature about the impact of IDS on the life time of the network. In this chapter, the affect of intrusion detection solutions on the lifetime of the wireless sensor networks is studied. With this new plug-in, it is possible not only to improve WSN system, in terms of security, but also at the same time to consider the over heads that may be caused and the effects on it on the WSNs life time. More specifically, comparisons between IDSs that continuously monitor the network and IDSs that use some kind of agreement in order to discover the attackers and isolate them is presented. The agreement considered in this approach is based on the Byzantine Generals solution introduced by

Lamport [Lamport, 1982]. With this new plug-in presented in this chapter, it is possible not only to improve WSN system, in terms of security, but also at the same time to consider the over heads that may be caused and the effects on it on the WSNs life time.

7.2 Related Work - The Byzantine's Problem

Intrusion detection is an important aspect within the broader area of network security, so an attempt to apply the idea in resource constrained WSN makes a lot of sense. Distributed systems are subject to a variety of failures and attacks. A survey study on intrusion detection systems is presented by Mishra et al. in [Mishra et al., 2004]. They identify security vulnerabilities in mobile ad-hoc networks and propose intrusion detection schemes. Sun et al. [Sun et al., 2007] presents a survey of intrusion detection techniques in mobile ad-hoc and wireless sensor networks. They also present existing solutions for secure localization and secure aggregation of data. Intrusion detection systems are predominantly classified as signature-based or anomaly-based. In the first approach, signatures containing typical attack characteristics or defined patterns are used to detect the attacks. On the other hand, anomaly-based detectors attempt to detect any type of deviations from the pre-defined profile of normal network behaviour. Signature based techniques are not capable of detecting new attacks; whereas anomaly based techniques can possibly detect new attacks. However, this advantage comes at the cost of possible false alarms, thus depleting the network life time.

The Byzantine problem, and appropriate solution approaches are popular especially for ad-hoc wireless networks. The problem of secure network communications in the presence of Byzantine problems has been extensively studied in [Dolev, 1981; Malki and Reiter, 1996; Papadimitratos and Haas, 2002]. Lamport et al. introduces the concept of Byzantine Generals Problem in [Lamport, 1982] and this is further developed by Dolev et al. in [Dolev et al., 1993]. Byzantine generals problem describes a problem where one

Commander and $n-1$ lieutenants communicate with each other. It is an abstraction of the problem of reaching an agreement in a system where the nodes may exhibit arbitrary behaviour. A Byzantine failure is defined as an arbitrary fault arising during the execution of an algorithm in a fault tolerant distributed computing system. Although a lot of emphasis is on secure networking for ad-hoc and sensor networks, the research work is still limited in distributed detection and data fusion in the presence of Byzantine problems [Shi and Perrig, 2004]. In [Kosut and Tong, 2008], Kosut et al. considered information theoretic investigation of data fusion in the presence of Byzantine problems. However, the authors were mainly interested in retrieving the data at the fusion centre and not in the detection performance.

The approach presented also improves the security of previous systems by recovering replicas pro-actively without necessarily identifying that they have failed or been attacked. This proactive recovery limits the time extent of a particular fault by regularly recovering replicas. In this way, the system works correctly even when all the replicas fail multiple times over the lifetime of the system, provided that less than one-third of the replicas are all faulty within a window of vulnerability.

Byzantine failure detectors provide an elegant abstraction for solving security problems. Without using cryptographic mechanisms, Byzantine problems can be tolerated by sending correct messages that outnumber the potential false messages [Bhandari and Vaidya, 2005; Koo et al., 2006; Pelc and Peleg, 2007; yuen Koo, 2004].

Lamport et al. [Lamport, 1982] have introduced an algorithm than can tolerate Byzantine failures. The application implements two Byzantine agreement protocols: Oral Message and Signed Message Algorithms.

Oral Message Algorithm: To cope with m traitors the authors proposed a solution that works for $3m+1$ or more lieutenants. The algorithm works in rounds where messages are exchanged between the lieutenants in each round.

Signed Messages Algorithm: Every lieutenant sends an un-forgeable signed message, preventing a traitor lieutenant from sending a value other than what he receives. The number of exchanged messages is minimized since

only unforgeable messages are sent. Byzantine fault tolerance techniques such as the state machine replication which can tolerate a bounded number of Byzantine faults can be used to protect the systems [Castro and Loskov, 1999]. Security issues in sensor networks are similar to ad-hoc networks but due to the energy restrictions of sensor networks, the defence mechanisms developed for ad-hoc cannot be directly applicable for sensor networks. Many ad-hoc network security mechanisms have been proposed for authentication and secured routing protocols in the literature based on the public key cryptography [Binkley and Trost, 2001; Hubaux et al., 2001; Kong et al., 2001, 2002; Luo et al., 2002; Sanzgiri et al., 2002; Zapata, 2002; Zhou and Haas, 1999]. Fewer secure routing protocols have been proposed based on the symmetric key cryptography [Basagni et al., 2001; Hu et al., a,b; Papadimitratos and Haas, 2002]. Cryptography is very expensive for sensor nodes [Karlof and Wagner, 2002]. The Byzantine Generals problem under various hypotheses can be used to implement reliable computer systems. However, these solutions are inherently very expensive in terms of both number of messages required and also the amount of time, especially when the fraction of faulty nodes is high.

A popular intrusion detection technique in WSNs is the *watchdog approach* [Mostarda and Navarra, 2008]. Each packet transmitted in the network is not only received by the sender and the receiver, but also from a set of neighbouring nodes within the senders radio range. Nodes use this information in order to detect anomalous behaviour. In other words in a watchdog approach nodes control with each other. This solution not only makes impossible to optimise the duty cycle (as the nodes can never sleep) but also requires the nodes to be in the same range. Furthermore promiscuous mode is not really suitable for applications with event-triggered sensing.

7.3 System Model - Assumptions

An WSN is composed of a set of nodes communicating by means of send and receive primitives. Nodes can be of different types such as temperature,

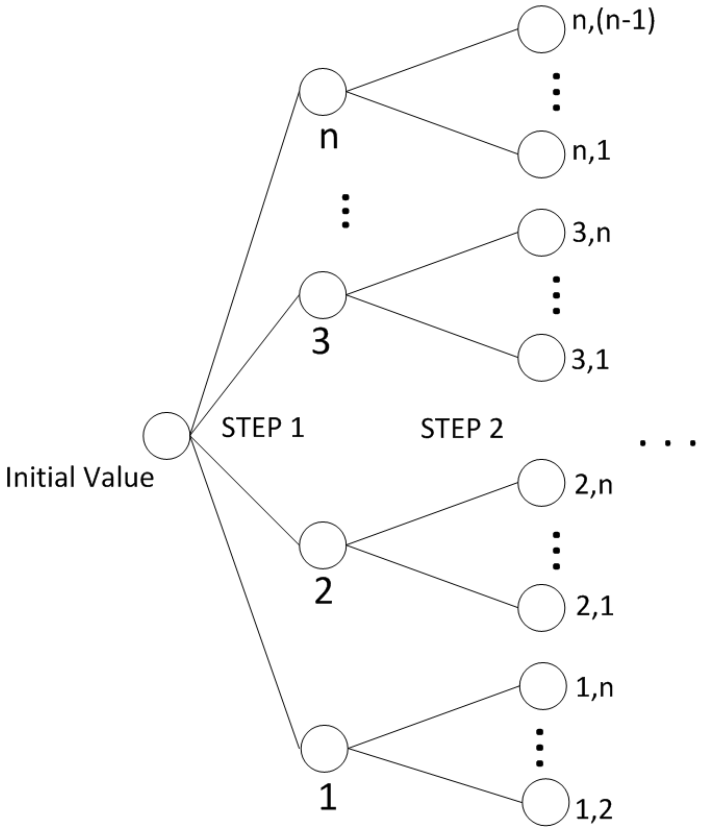

Figure 7.1: The tree layout for n number of nodes

smoke and sprinkler. Each type of sensor can have several instances. Each node has a unique address and the possibility of the implementation of a transport layer protocol, allowing end-to-end communication between node. In this work, unreliable links and unpredictable delays for the wireless links are considered, and also assume each node has a public and private key pair that can be used to sign messages. The attacker can physically compromise one or more nodes. Once the attacker has compromised a node it can install any code and it can use node's credentials to send signed messages. A compromised node does not follow the protocol and can send malicious messages in order to subvert the correct system behaviour. For instance, a malicious node does not allow the detection of a fire or it could enable the water flow of the sprinklers when no fire is present. In this work, assumption that the communication between two honest nodes always exists, and there is secure communication path between an honest node and the base station is considered. This is used to deliver alert messages. Another assumption is that the attacker does not compromise all the sensor nodes and majority of the nodes are honest i.e. the correct protocol is followed. To help with the understanding, the simulation is carried upto a 100 nodes. In the first stage, which is simply a data gathering stage, the algorithm defines m+1 rounds of messaging between all the processes. It's easy to see that as the number of processes increases, the number of messages being exchanged starts to go up rapidly. If there are N processes, each process sends N-1 messages in round 1, then (N-1)*(N-2) in round 2, (N-1)*(N-2)*(N-3) in round 3. That can add up to a lot of messages in a big system causing delays. Say if there are 6 processes, each process sends 5 messages in round 1, then 20 in round 2, and 60 in round 3. The figure presented in 7.1 represents the case study considered in this study.

7.3.1 Case Study

The case study considered is a home monitoring system of building environment which is used for WSNs quite often. Please note that, the case study chosen is a typical example of a WSN system where the sensors are event

triggered, or the time between observations is relatively long (in other words the number of packets exchanged between the base station and the nodes is spread in a long time period). For these kind of applications, although the number of messages exchanged for establishment of communication is not a real burden, in case of external attacks, the use of IDSs can be the main cause of energy loss and processing delay.

Home monitoring systems include emergency control systems (e.g. fire alarms). The fire alarm system is composed of different temperature sensors and smoke detectors that are distributed uniformly inside the building. There are also sprinkler actuators used to enable the water flow in case of fire. When a temperature sensor reads a value that exceeds a specified threshold; it sends an alert message to the smoke detector. The smoke detector receives the alert and checks for smoke. An alarm is raised when the smoke is detected. In this case the smoke sensor also activates all the sprinklers. The life time of the fire alarm system is evaluated when intrusion detection facilities are introduced, the following scenarios are simulated:

- a fire alarm system without any IDS facility
- a fire alarm system in which an agreement based on Byzantine protocol is used to detect and isolate the attacker
- a fire alarm system in which the IDS is distributed on each node and runs in promiscuous mode that the IDS component continuously eavesdrop the network

The following simulation parameters are used: CC2420 radio defined by the Texas instruments is used. The packet rate is kept at 250 kbps, the radio bandwidth is 20 MHz and the simulation is run for 9000 sec. The nodes are deployed uniformly across the area as shown in figure 7.2.

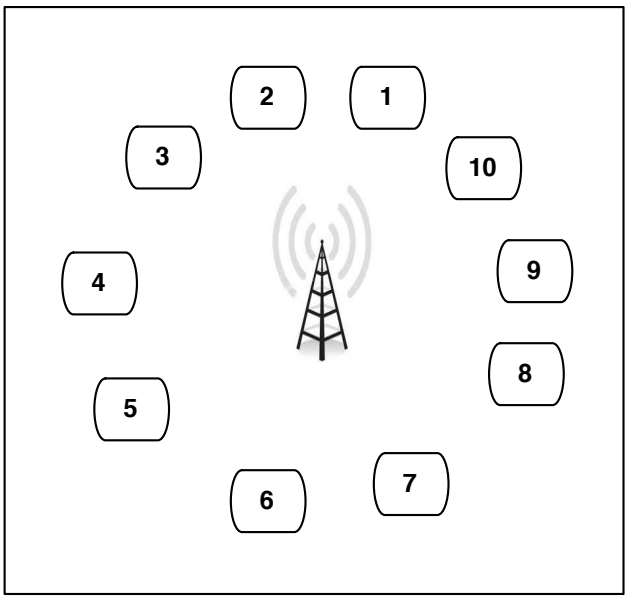

Figure 7.2: Case study considered

7.4 Numerical Results

With the help of the new plug-in (placeLife) developed for performance evaluation of WSNs, it is possible not only to improve WSN system, in terms of security, but also at the same time to consider the over heads that might be caused and the effects on the life time of the WSN. In this section numerical results are presented for the fire alarm system. Results are obtained to check the cost of two different intrusion detection systems. The first system is watchdog based approach in which the IDS is distributed in each node and the radio is assumed to be always on so that the IDS can capture and analyse all the messages. The second IDS uses the solution of Byzantine generals problem using oral messages solution as offered in [Lamport, 1982] to detect and isolate the attacker. In order to demonstrate the cost of the considered IDSs more evidently, numerical results are provided for a system without IDSs as well.

Castalia simulation package is used which considers the stochastic processes; job arrivals and departures, in an event triggered fashion. Although the

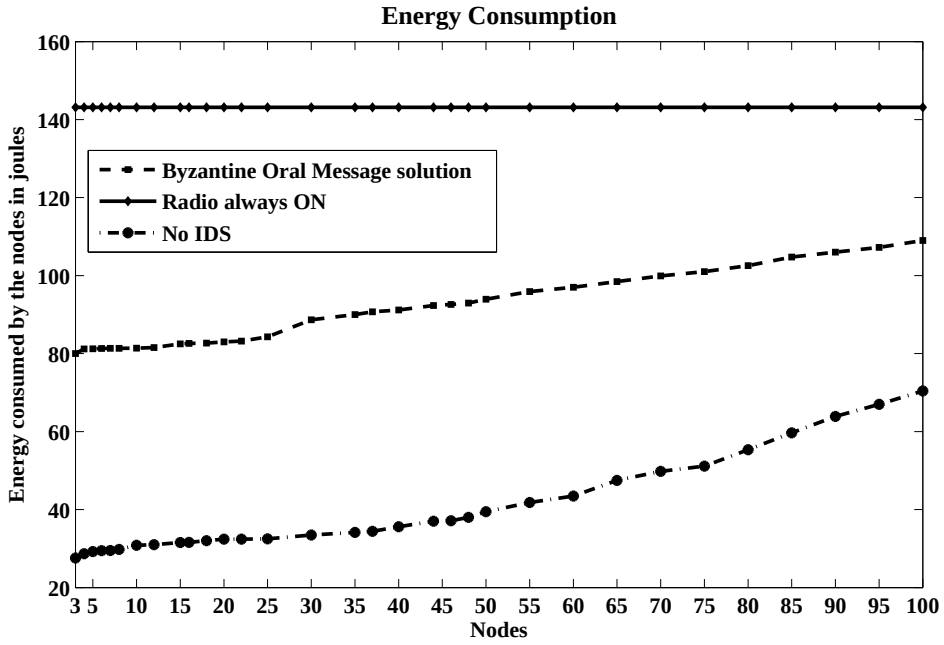

Figure 7.3: Energy consumed for the IDSs as a function of number of nodes

OMNET package is event triggered, the simulation time was quite extensive especially for the scenarios with more than 70 nodes. Results presented in figure 7.3 are given for up to 100 nodes. In this approach, 60% and 30% of the nodes are assumed to be sensors of temperature and smoke, respectively, while 10% of the nodes are assumed to be sprinkler actuators. For all the scenarios the simulation is run for 9000 seconds.

The lowest line of Figure 7.3 shows the energy consumed by a single sensor of temperature when no IDS is introduced. More specifically the temperature node is sensing the temperature continuously and sending a reading to the smoke sensor every 30 seconds. The average energy consumed is 30.82, 39.462 and 70.397 joules for each node in a wireless sensor network composed of 10, 50 and 100 nodes, respectively.

The middle line of Figure 7.3 shows the energy consumed by a temperature node that runs for 9000 sec and performs exactly one instance of the Byzantine agreement. The temperature node is assumed to sense for the temperature, and after it runs the Byzantine agreement. This is run with all

remaining temperature nodes in order to understand whether or not the temperature is indeed high or some of the nodes is trying to attack the system. After running the Byzantine agreement the temperature node continues to sense temperature and sends a message to all the smoke sensors every 30 seconds. The average energy consumed is 81.389, 93.92 and 109.004 joules for each of the node in wireless sensor network of 10, 50 and 100 nodes, respectively.

The topmost line of Figure 7.3 shows the energy consumed by a node that has the radio receiver always on that is 143.125 joules. This is the case in which the node is hosting an IDS component that is continuously eavesdropping the network.

For energy calculations Byzantine protocol is considered to see the impact of security mechanisms on energy consumption in WSNs. The results presented clearly show that the energy consumed because of the Byzantine protocol increases as the number of nodes in the system increases. Although, it was expected to have rapid increases for the energy consumed is almost linear. The main reason of having this behaviour is that there is an upper-bound for the maximum energy that can be consumed by each node. The maximum energy that can be consumed is equal to the energy consumed when the radio is always on. Since for a Byzantine system with n nodes the number of messages delivered can be computed as $(n-1)$ for the first step $(n-1)(n-2)$ for the second step and $(n-1)(n-2)\dots(n-k)$ for step k, for systems with large numbers of nodes, the radio is always kept on in order to deal with incoming and outgoing packets.

The delay introduced by the intrusion detection system should also be considered. According to the authors [Russello et al., 2011], the average time in order to verify a rule for intrusion detection is about about 2.8 ms. It means that when the IDS is running in promiscuous mode and the sensor node is not overloaded with messages from its neighbours, the node should be able to perform all sensing and detection activities. When the Byzantine agreement is considered the delay before the node restarts its reading and sensing activities can be quite long. There exists an inherent energy-

accuracy-latency-security trade-off present in sensor networks. That is, the more energy one is willing to give, the more accurate, less latency and more security is achieved, or by keeping the energy consumption constant, one can trade high accuracy for lower latency and un-secure connections. The parameters of the scheme are determined as a trade-off between security in the case of compromised nodes and the probability that the network is connected. This improved the protection of the network against small-scale networks which are easy to execute and therefore more likely, and decreases the protection against larger attacks, which are more expensive to perform.

7.4.1 Summary and Recommendations

Like every other computer network, wireless sensor networks are exposed to a variety of threats and attacks and like most other networks, sensor networks require support for confidentiality, integrity, and authentication to protect sensor nodes and sensor data. In this chapter, the cost of intrusion detection systems in the context of WSNs is considered. Although there are a number of security measures offered, and implemented for WSNs and/or wireless ad-hoc networks, the discussions of performance of the implemented algorithm should go beyond how secure the network becomes and what kind of attacks can be detected etc. Especially in case of WSNs, the matter becomes very delicate since the energy consumption can have severe effects on the network lifetime while the network is being protected. The impact on the energy consumption when intrusion detection systems are employed together with wireless sensor networks is critically evaluated.

A novel intrusion detection system should not eavesdrop the network all the time nor should run a Byzantine agreement involving too many nodes. The former solution would deplete the energy of the node quite quickly while the latter would require each node to run the agreement for a long time suspending the sensing and reading activities.

An intrusion detection system for wireless sensor networks should have the following characteristics:

- it should not run in promiscuous mode all the time;
- it should locally check the information on each node and start interaction with other nodes when a suspicious activity is detected;
- the detection of an attack (after detecting a suspicious activity) should require agreement between few nodes without involving all nodes of the WSN;

It is believed that, most of the time the attack is localised on some part of the network. There should be an agreement to allow only those nodes should in the detection of the attacks. Byzantine solution based on signature can also be explored. Although this can reduce the number of messages, since computing the signature for a sensor node can be quite energy consuming. With the help of PlaceLife developed for performance evaluation (in terms of energy consumed, life time of the network, delay incurred, packets lost, packets delivered) of WSNs, it is possible not only to improve WSN system, in terms of security, but also at the same time to consider the over heads that might be caused and the effects on the life time of the WSN. Numerical results presented give the effectiveness of the tool enabling the need to consider the collaborative nature of WSNs and their correlative characteristics, considering issues from physical to application layer together, to enable the framework designed to provide WSN lifetime estimation and performance evaluation in terms of improving the performance and maximizing the lifetime of the network. As WSNs continue to become more commonplace, it is to expect that security challenges will increase, the types and number of threats will evolve, and new solutions to protect sensor networks and sensor data will be required.

The ultimate security objective is to provide confidentiality, integrity, authenticity, and availability of all messages in the presence of resourceful adversaries. Every eligible receiver should receive all messages intended for it and be able to verify the integrity of every message as well as the identity of the sender. Developing a security approach should consider the capacities of resources (memory, processor, and power supply) of wireless sensor nodes.

It is an expected result that additional encryption mechanisms to increase security in WSN applications increase the node power consumption amounts and the average end-to-end delay times. This has been also confirmed from the results presented in this chapter. Hence, it is of great importance to determine the requirements of the application very well. In large-scale or industrial WSN applications, security is not so important, whereas power consumption is very significant. On the other hand, security is very crucial in military and health care applications while power consumption can be relatively ignored. For this very reason, it is always important to select an encryption algorithm and an encryption mode appropriate for the security solutions developed to be used in military and health care applications.

The security solutions that are developed should be modular, i.e., new encryption algorithms and modes developed are to be better in terms of security, power consumption, memory usage, and delay issues, and they must also be able to be integrated into the developed security solution directly. It is of utmost necessity to develop a security solution which tunes with every aspect of the security requirements (data privacy, data integrity, data freshness, identity authentication, and availability) of WSN, while considering the idea of high security and low power consumption for each requirement. Also, it is a drawback for researches that most of the recommended security solutions remained in the simulation environment, and they are not tried on sensor platforms. Hence, it is also necessary for the recommended protocols to be used directly in applications requiring security. When developing a security solution, the most appropriate one must be selected by taking into consideration the WSN characteristics, the security requirements, the attacks, and the current encryption algorithms and modes.

Chapter 8

Conclusion

8.1 Contributions of the Thesis

The vast number of solutions that have driven the research community over the years have made WSN phenomenon a reality, and hence are recognized as one of the most important technologies for the twenty-first century. It is envisioned that in the near future WSNs will be widely used in various civilian and military fields, and revolutionize the way we live, work, and interact with the physical world. MIT's Technology Review Magazine called wireless sensor networks as one of the ten technologies that will change the world. Sensor networks offer countless challenges, but their versatility and their broad range of applications are eliciting more and more interest from the research community as well as from industry. Sensor networks have the potential of triggering the next revolution in information technology.

In this research project, an approach is presented considering the collaborative nature of WSNs and its correlation characteristics, also providing a tool which considers issues from physical to application layer together as entities to enable the A4WSN framework. The simulation approach considered considers careful assumptions for various layers, also providing a clear separation of concerns amongst software architecture of the applications, the hardware configuration and the WSN deployment unlike the existing tools for evalua-

tion. The reuse of models across projects and organizations is also promoted while realistic WSN lifetime estimations and performance evaluations are possible in attempts of improving performance and maximizing the lifetime of the network. The case studies presented demonstrate the importance of various parameters considered in this study. Simulation-based studies are presented for all the issues considered. The performance of the layered protocol stack in realistic settings reveals several important interactions between different layers. These interactions are especially important for the design of WSNs in terms of providing realistic estimations and performance evaluation and for maximizing the lifetime of the network. The objectives listed in the introduction chapter have been achieved and the details are presented below. The main focus of the thesis was to evaluate WSNs in a more realistic way. The objectives include:

- The main contribution of the thesis, an approach considering the collaborative nature and correlation characteristics of WSNs is presented.
- The tool presented enables the presented framework by considering issues from physical to application layer together as entities.
- The simulations were carried out with real time characteristics of WSNs and careful assumptions for various layers are taking into account.
- The framework presented provided a clear separation of concerns amongst software architecture of the application, the hardware components and the WSN deployment considered.
- These interactions are especially important for the design of WSNs in terms of realistic life time estimations and performance evaluations and for maximising the lifetime of the network, as maximizing the lifetime is an important goal in sensor networks.
- In WSNs, the common configurations is to prolong the lifetime and deal with the path loss phenomena by having a multi-hop set-up with clusters and cluster heads to relay the information. Although researchers

continue to address these challenges, the type of data arrival distributions at the cluster head and intermediary routing nodes remains to be an interesting area of investigation. In published works, the general practice is to compare an empirical exponential arrival distribution of wireless sensor networks with a theoretical exponential distribution in a Q-Q plot diagram. In chapter 3, we show that such comparisons based on simple eye checks are not sufficient since, in many cases, incorrect conclusions may be drawn from such plots. After estimating the Maximum Likelihood parameters of empirical distributions, we generate theoretical distributions based on the estimated parameters. To the best of our knowledge, this is the first work that provides statistical proof for finding theoretical distributions of arrivals at the CH and relay nodes in WSNs. Kolmogorov-Smirnov Test Statistics are conducted for each generated inter-arrival time distributions in order to find out, if it is possible to represent the traffic into the cluster head by using theoretical distribution. In wireless sensor networks empirical exponential arrival distribution assumption holds only for a few cases. There are both theoretically known such as Gamma, Log-normal and Mixed Log- Normal of arrival distributions and theoretically unknown such as non-Exponential and Mixed cases of arrival distributions in wireless sensor networks. The work is further extended to understand the effect of delay on inter-arrival time distributions based on the type of medium access control used in wireless sensor networks.

– Popularity of cloud computing is increasing day by day in distributed computing environment. There is a growing trend of using cloud environments for storage and data processing needs. Cloud computing provides applications, platforms, and infrastructure over the Internet. It is a new era of referring to access shared computing resources. On the other hand, wireless sensor networks have been seen as one of the most essential technologies for the 21st century where distributed spatially connected sensor nodes automatically forms a network for data transmission. WSNs development is still plagued by many issues despite the

ever increasing usage of WSNs in modern applications. These issues are to be addressed, hence ameliorating the implementation and enabling different analysis in terms of various factors, such as energy efficiency and performance evaluation to be performed at the early stages of WSN design and development. A rich multi-view modelling environment has been proposed, supported by a powerful programming framework, for the model-driven engineering of wireless sensor networks in Chapter 4. The modelling viewpoints and conceptual elements have been carefully designed, including the domains such as software engineering, wireless sensor networks, simulations and telecommunications. The programming frameworks functioning has been tested by realizing a plug-in devoted to energy and performance related simulations of WSNs.

– WSNs are comprised of small, inexpensive sensors with wireless communication capabilities, called motes. They are deployed in adhoc networks and are powered by limited power supplies. These motes are deployed in large numbers and provide unprecedented opportunities for instrumenting and controlling homes, cities and the environment. They find applications in different fields like military sensing, physical security, air traffic control, traffic monitoring, video surveillance, industrial automation etc. Each poses different challenges for these motes but one common challenge faced in all fields is power conservation. This is because motes are sometimes deployed in difficult to reach regions and this makes it difficult to replace the batteries. Hence power conservation becomes an important factor for these motes. One of the main reasons for deploying these motes in adhoc is power conservation. Power is consumed during data processing and RF communication, but communication electronics uses far more power than processing. In order to validate the expressivity of the A4WSN modelling languages and to exercise the provided extension points, an analysis plug-in called PlaceLife has been developed, and is presented in Chapter 5. An overview on the existing simulators is also presented. This work shows that when path loss is introduced, increasing the transmission

power is needed to reduce the amount of lost wickets. This presents a trade-off between the residual energy and the successful transmission rate when more realistic settings are employed for simulation. In order to show the usefulness and effectiveness of the approach presented, a case study based on home automation system is also presented. Numerical results along with the analysis of various factors affecting the performance in terms of energy consumption of WSNs are given.

– The large-scale deployment of wireless sensor networks (WSNs) and the need for data aggregation necessitate efficient organization of the network topology for the purpose of balancing the load and prolonging the network lifetime. Clustering has proven to be an effective approach for organizing the network into a connected hierarchy. Since it is likely that the data acquired from one sensor node is highly correlated. Thus, node clustering, which aggregates nodes into groups (clusters), is critical to facilitate practical deployment and operation of WSNs. In Chapter 6, the major issues and challenges in node clustering for WSNs are discussed and a variety of state-of-the-art clustering techniques are introduced and discussed in detail. The affect of path loss on well known clustering protocols are also presented. Also, the bottlenecks in the network, in terms of cluster size scalability, especially while addressing variety of high packet sending rate and real-time applications, such as wearable heart rate and physical activity monitors and holster monitors is presented.

– Though sensor networks share some commonalities with typical computer networks, the unique requirements of its own make them a special type of network. They consist of sensor nodes deployed in a manner to collect information about surrounding environment. Their distributed nature, multihop data forwarding, and open wireless medium are the factors that make WSNs highly vulnerable to security attacks at various levels. Intrusion Detection Systems (IDSs) can play an important role in detecting and preventing security attacks Wireless sensor networks pose new security challenges because of their unattended nature

and limited resources. Intrusion detection in WSN is a particularly challenging task because of the limited resources of the nodes. WSNs can operate in two different modes called as continuous periodic sensing and transmission or event-triggered sensing. The decision on which mode of operation to use is highly dependant on the application. In Chapter 7, the affects of intrusion detection solutions on the lifetime of the WSNs is studied. More specifically, comparisons between approaches that continuously monitor the network and those that use some kind of agreement in order to discover the attackers and isolate them is presented.

8.2 Future Directions

WSNs are increasingly gaining impact in our day to day lives. They are finding a wide range of applications in various domains and are playing a key role in several application scenarios such as healthcare, agriculture, environment monitoring, and smart metering. The Internet is smoothly migrating from an Internet of people towards an Internet of Things. It is commonly accepted that the next generation of internet is becoming the "Internet of Things (IoT)" which is a worldwide network of interconnected objects and their virtual representations uniquely addressable based on standard communication protocols. As a result, WSNs are an invaluable resource for realizing the vision of the IoT [Akyildiz and Vuran, 2010; Alcaraz et al., 2010; Langendoen et al., 2014]. In order to encompass the applicability of WSN architecture and to provide useful information anytime and anywhere, it is of crucial importance to integrate with the Internet. Realization of these networks will require tight integration and interoperability; however, so far, research has progressed in each of these areas separately. Therefore, it is of crucial significance to develop energy efficient, location and spectrum-aware cross-layer communication protocols as well as heterogeneous network management tools for the integration of WSNs, cognitive radio networks, mesh networks, and the Internet. Rapid technological advancement in wire-

less sensor networks has contributed for the great increase in number of IP-connected smaller smart sensors that, in turn, become part of the IoT. Thus, the interconnection of wireless sensor devices to the Internet facilitates the M2M communication with many application areas such as smart grid, metering, health, environment, vehicle and home appliances. The innovations of integration of WSNs into iThings offer many interesting avenues of research for scientific communities. The research into WSNs for IoT is extremely important which can possibly change our day-to-day lives. The key for IoT applications are the ability to interact with physical world through computation, communication, and machine control. However, each sensor device in IoT cannot conveniently communicate with other terminal devices through internet protocol. So, it is ideal to establish protocol translation stack or equipment between two WSN groups. The development of 6LoWPAN standard to integrate the IPv6 standard with low-power sensor nodes led to efficient integration for communication between IPv6-based devices and sensor motes. However, significant challenges still exist at higher layers of the protocol stack in seamless integration between WSNs and the Internet.

The Third Industrial Revolution is picking up speed as "Open standards in the IoT are challenging closed platform approaches, while wearable and other consumer and lifestyle technology are opening up new ecosystems and opportunities". Sensing is a very important to the construction of IoT. Multiple sensors with communication and computing power are connected in wireless and cooperate with each other to exchange with physical world and accomplish specific application tasks, forming a sensor network. Sensor network integrates various inter disciplinary technologies, such as sensor technology, embedded computing technology. A multi-hop self-organizing system is established in the sensor network through wireless communication, which is responsible for perception collection, processing information perceived within the coverage area of the network, allowing flexible monitoring of the environment. To achieve this vision, there is a need for scalable and interoperable networking systems to support the challenging requirements for future internet and web. The challenges include security, reliability, energy-efficient and

cost-effective large-scale sensor networks, machine-to-machine communications, and information networking architectures that are suitable for low end devices through to high end consumers. The study presents an insight on how a generalized framework considering the collaborative nature of WSNs and its correlation characteristics, is necessary for data collection and modelling, that effectively exploits spatial and temporal characteristics of the data, both in the sensing domain as well as the associated transform domains, also providing the impact of realistic approach for evaluating WSNs. The combination of these factors has improved the visibility of utilizing a sensor network consisting of intelligent sensors, enabling data collection, processing and analysing the gathered information in a variety of environments, making ways to take full advantage of the available internet technology. Such consolidation is quite clearly accelerating progress towards an IoT, providing an overarching view for the integration and functional elements that can deliver an operational IoT.

Chapter 9

Appendix A

Some Probability Distributions

Exponential Distribution

This is a distribution of the time to an event when the probability of the event occurring in the next small time interval does not vary through time. It is also the distribution of the time between events when the number of events in any time interval has a Poisson distribution. The exponential distribution is characterized as follows:

Definition Let X be an absolutely continuous random variable. Let its support be the set of positive real numbers:

$$R_X = [0, \infty) \tag{9.1}$$

Let $\lambda \in R_{++}$. We say that X has an exponential distribution with parameter λ (rate parameter) if its probability density function is:

$$f_X(x) = \begin{cases} \lambda exp(-\lambda x) & \text{if } x \in R_X \\ 0 & \text{if } x \notin R_X \end{cases}$$

The exponential distribution has many applications. Examples include the time to decay of a radioactive atom and the time to failure of components with constant failure rates. It is used in the theory of waiting lines or queues, which are found in many situations: from the gates at the entrance to toll roads through the time taken for an answer to a telephone enquiry, to the time taken for an ambulance to arrive at the scene of an accident. For exponentially distributed times, there will be many short times, fewer longer times, and occasional very long times.

The exponential distribution is also known as the negative exponential distribution. The exponential distribution is the only continuous distribution characterized by a "lack of memory".

Gamma Distribution

The gamma distribution includes the chi-squared, Erlang, and exponential distributions as special cases, but the shape parameter of the gamma is not confined to integer values. The gamma distribution starts at the origin and has a flexible shape.

Log-normal Distribution

In probability theory, a log-normal distribution is a probability distribution of a random variable whose logarithm is normally distributed. If Y is a random variable with a normal distribution, then $X = \exp(Y)$has a log-normal distribution; likewise, if X is log-normally distributed, then $Y = \log(X)$is normally distributed. The log-normal distribution is applicable to random variables that are constrained by zero but have a few very large values. The resulting distribution is asymmetrical and positively skewed.

The application of a logarithmic transformation to the data can allow the data to be approximated by the symmetrical normal distribution, although the absence of negative values may limit the validity of this procedure.

Mixture Distributions

A mixture distribution has a distribution function with a representation as a convex combination of other specific probability distribution functions. A mixture may be comprised of a finite number of base elements, where usually a relatively small number of individual distributions are combined together, or an infinite number of base elements. Often an individual base distribution is thought of as representing a unique sub population within the larger (sampled) population. In both the finite and infinite case, the probability of an outcome may be thought of as a weighted average of the conditional probabilities of that outcome given each base distribution, where the relevant mixture weight describes the relative likelihood of a draw from that distribution being obtained.

Finite Mixture

A *finite mixture of two distributions* having cdfs $F_1(x)$ and $F_2(x)$, respectively, has cdf $F_x = \eta F_1(x) + (1 - \eta)F_2(x)$, as long as $0 < \eta < 1$. Extending this notion to a finite mixture of K distributions (sometimes referred to as a finite K-mixture) involves using a convex combination of distinct distribution functions. As the combination is convex, each of the mixture weights $\eta_1, \eta_2, \ldots, \eta_k$ are between zero and one, and sum to unity.

Due to their ability to combine very different distributional structures, finite mixture distributions are well suited to cater for a large range of empirical distributions in practice. However, finite mixture models are often over-parametrized, leading to identification issues.

Appendix B

Probability Plots or Quantile-Quantile Plots

A probability plot or quantile-quantile (Q-Q) plot is a graphical display invented by Wilk and Gnanadesikan [Wilk and Gnanadesikan, 1968], to compare a data set to a particular probability distribution or to compare it to another data set. The idea is that if two population distributions are exactly the same, then they have the same quantiles (percentiles), so a plot of the quantiles for the first distribution versus the quantiles for the second distribution will fall on the $0-1$ line (i.e., the straight line $y = x$ with intercept 0 and slope 1). If the two distributions have the same shape and spread but different locations, then the plot of the quantiles will fall on the line $y = a + x$ (parallel to the $0-1$ line) where a denotes the difference in locations. If the distributions have different locations and differ by a multiplicative constant b, then the plot of the quantiles will fall on the line $y = a + bx$ [Hirsch et al., 1991; Millard et al., 2000]. Various kinds of differences between distributions will yield various kinds of deviations from a straight line.

References

Opnet, December 2011a. URL http://www.opnet.com. 37, 101

Castalia, December 2011b. URL http://castalia.npc.nicta.com.au. 119

Qualnet, December 2011c. URL http://www.scalable-networks.com. 37, 101

Sinalgo, December 2011d. URL http://dcg.ethz.ch/projects/sinalgo. 37, 100

Shawnwiki, January 2012. URL http://shawnwiki.coalesenses.com/index.php/Shawn_Introduction. 37, 100

Corie, 2013. URL http://www.ccalmr.ogi.edu/corie. 23

Ameer Ahmed Abbasi and Mohamed Younis. A survey on clustering algorithms for wireless sensor networks. *Comput. Commun.*, 30(14-15):2826–2841, October 2007. ISSN 0140-3664. 6

Kemal Akkaya and Mohamed Younis. A survey on routing protocols for wireless sensor networks. *Ad Hoc Networks*, 3:325–349, 2005. 35

I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. Wireless sensor networks: a survey. *Computer Networks*, 38:393–422, 2002a. 2, 16, 21, 22, 32

Ian F Akyildiz, Weilian Su, Yogesh Sankarasubramaniam, and Erdal Cayirci. Wireless sensor networks: a survey. *Computer networks*, 38(4):393–422, 2002b. 40

I.F. Akyildiz and M.C. Vuran. *Wireless Sensor Networks*. Advanced Texts in Communications and Networking. Wiley, 2010. ISBN 9780470515198. URL http://books.google.co.uk/books?id=7YBHYJsSmS8C. 2, 7, 15, 16, 17, 23, 24, 32, 170

I.F. Akyildiz, Su Weilian, Y. Sankarasubramaniam, and E. Cayirci. A survey on sensor networks. *Communications Magazine, IEEE*, 40(8):102 – 114, aug 2002c. ISSN 0163-6804. 3, 13

Jamal N. Al-karaki and Ahmed E. Kamal. Routing techniques in wireless sensor networks: A survey. *IEEE Wireless Communications*, 11:6–28, 2004. 16

Cristina Alcaraz, Pablo Najera, Javier Lopez, and Rodrigo Roman. Wireless sensor networks and the internet of things: Do we need a complete integration? In *1st International Workshop on the Security of the Internet of Things (SecIoT'10)*, Tokyo (Japan), December 2010. IEEE. 170

Hande Alemdar and Cem Ersoy. Wireless sensor networks for healthcare: A survey. *Comput. Netw.*, 54(15):2688–2710, October 2010. ISSN 1389-1286. 23, 88, 91

Majd Alwan, Jon Leachtenauer, Siddharth Dalal, David Mack, Steve Kell, and Beverely Turner. Impact of monitoring technology in assisted living: Outcome pilot. *IEEE Transactions on Information Technology in Biomedicine*, 10, 2006. 22, 23, 88

Th. Arampatzis, J. Lygeros, Senior Member, and S. Manesis. A survey of applications of wireless sensors and wireless sensor networks. In *Proc. 13 th Mediterranean Conference on Control and Automation, Limassol*, pages 719–724, 2005. 21

J. Banks. *Discrete-event System Simulation*. Prentice Hall PTR, 2010. ISBN 9780136062127. URL http://books.google.co.uk/books?id=cqSNnmrqqbQC. 3

Stefano Basagni, Kris Herrin, Danilo Bruschi, and Emilia Rosti. Secure pebblenets. In *Proceedings of the 2nd ACM international symposium on Mobile ad hoc networking & computing*, MobiHoc '01, pages 156–163, New York, NY, USA, 2001. ACM. ISBN 1-58113-428-2. 156

Elizabeth A. Basha, Sai Ravela, and Daniela Rus. Model-based monitoring for early warning flood detection. In *Proceedings of the 6th ACM conference on Embedded network sensor systems*, SenSys '08, pages 295–308, New York, NY, USA, 2008. ACM. ISBN 978-1-59593-990-6. 23

H. Ben Chikha, A. Makhlouf, and W. Ghazel. Performance analysis of aodv and dsr routing protocols for ieee 802.15.4/zigbee. In *International Conference on Communications, Computing and Control Applications (CCCA)*, pages 1 –5, March 2011. doi: 10.1109/CCCA.2011.6031459. 102

Yann Ben Maissa, Fabrice Kordon, Salma Mouline, and Yann Thierry-Mieg. Modeling and analyzing wireless sensor networks with VeriSensor. pages 60–76, 2012. 39

Vartika Bhandari and Nitin H. Vaidya. On reliable broadcast in a radio network. In *Proceedings of the twenty-fourth annual ACM symposium on Principles of distributed computing*, PODC '05, pages 138–147, New York, NY, USA, 2005. ACM. ISBN 1-58113-994-2. 155

G. Bianchi. Performance analysis of the ieee 802.11 distributed coordination function. *IEEE J.Sel. A. Commun.*, 18(3):535–547, September 2006. ISSN 0733-8716. 44

Jim Binkley and William Trost. Authenticated ad hoc routing at the link layer for mobile systems. *Wirel. Netw.*, 7(2):139–145, March 2001. ISSN 1022-0038. 156

J. Blumenthal, M. Handy, F. Golatowski, M. Haase, and D. Timmermann. Wireless sensor networks - new challenges in software engineering. In *Emerging Technologies and Factory Automation Proceedings. ETFA '03. IEEE Conference*, volume 1, pages 551–556 vol.1, 2003. 36, 72

P. Bonnet, J. Gehrke, and P. Seshadri. Querying the physical world. *Personal Communications, IEEE*, 7(5):10–15, 2000. ISSN 1070-9916. doi: 10.1109/98.878531. 23

Nizar Bouabdallah, Mario E Rivero-Angeles, and Bruno Sericola. Continuous monitoring using event-driven reporting for cluster-based wireless sensor networks. *Vehicular Technology, IEEE Transactions on*, 58(7):3460–3479, 2009. 40

Miguel Castro and Barbara Loskov. Practical byzantine fault tolerance, February 1999. 156

Alberto Cerpa, Jeremy Elson, Deborah Estrin, Lewis Girod, Michael Hamilton, and Jerry Zhao. Habitat monitoring: application driver for wireless communications technology. *SIGCOMM Comput. Commun. Rev.*, 31(2 supplement):20–41, April 2001. ISSN 0146-4833. 23

Ruay Shiung Chang and Chia Jou Kuo. An energy efficient routing mechanism for wireless sensor networks. In *20th International Conference onAdvanced Information Networking and Applications, 2006. AINA*, volume 2. 147

Z. Che-Aron, W.F.M. Al-Khateeb, and F. Anwar. The enhanced fault-tolerant aodv routing protocol for wireless sensor network. In *Second International Conference on Computer Research and Development*, pages 105 –109, may 2010. doi: 10.1109/ICCRD.2010.74. 102

Gilbert Chen, Joel Branch, Michael J. Pflug, Lijuan Zhu, and Boleslaw K. Szymanski. Sense: A wireless sensor network simulator. 3, 101

Haiguang Chen, Peng Han, Xi Zhou, and Chuanshan Gao. Lightweight anomaly intrusion detection in wireless sensor networks. In *Intelligence and*

Security Informatics, volume 4430 of *Lecture Notes in Computer Science*, pages 105–116. 2007. 10.1007/978-3-540-71549-8-9. 28, 153

Yingying Chen, Jie Yang, W. Trappe, and R.P. Martin. Detecting and localizing identity-based attacks in wireless and sensor networks. *Vehicular Technology, IEEE Transactions on*, 59(5):2418–2434, Jun 2010. ISSN 0018-9545. doi: 10.1109/TVT.2010.2044904. 34

C. Chiasserini and M. Garetto. An analytical model for wireless sensor networks with sleeping nodes. *IEEE Transactions on Mobile Computing*, 5(12):1706–1718, Dec 2006. 41

C.F. Chiasserini and M. Garetto. Modeling the performance of wireless sensor networks. In *INFOCOM, Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies*, volume 1, pages 4 vol. (xxxv+2866), march 2004. doi: 10.1109/INFCOM.2004.1354496. 40, 41, 43, 48, 144

Octav Chipara, Christopher Brooks, Sangeeta Bhattacharya, Chenyang Lu, Roger D. Chamberlain, Gruia catalin Roman, and Thomas C. Bailey. Reliable real-time clinical monitoring using sensor network technology. 91

Krzysztof Czarnecki and Michal Antkiewicz. Mapping features to models: A template approach based on superimposed variants. In *GPCE*, pages 422–437, 2005. 82

W. Dargie and C. Poellabauer. *Fundamentals of Wireless Sensor Networks: Theory and Practice*. Wireless Communications and Mobile Computing. Wiley, 2010. ISBN 9780470975688. URL http://books.google.co.uk/books?id=8c6k0EVr6rMC. 152

Didonet Del Fabro M., Bézivin J., Jouault F. and Breton E. and Gueltas G. AMW: a generic model weaver. In *Proc. of 1Ãĺre JournÃľe sur l'IngÃľnierie DirigÃľe par les ModÃĺles, Paris, France. pp 105-114*, 2005. 82

Danny Dolev. The byzantine generals strike again. Technical report, Stanford, CA, USA, 1981. 154

Danny Dolev, Cynthia Dwork, Orli Waarts, and Moti Yung. Perfectly secure message transmission. *J. ACM*, 40(1):17–47, January 1993. ISSN 0004-5411. 154

Wan Du, Fabien Mieyeville, David Navarro, Ian OâĂŹConnor, and Laurent Carrel. Modeling and simulation of networked low-power embedded systems: a taxonomy. *EURASIP Journal on Wireless Communications and Networking*, 2014(1):106, 2014. doi: 10.1186/1687-1499-2014-106. URL http://dx.doi.org/10.1186/1687-1499-2014-106. 99

I.M.M.E. Emary and S. Ramakrishnan. *Wireless Sensor Networks: From Theory to Applications*. Telecommunications books. Taylor & Francis, 2013. ISBN 9781466518100. URL http://books.google.co.uk/books?id=3ad7AAAAQBAJ. 2, 44, 45, 54

Arati Manjeshwar et.al. Apteen : a hybrid protocol for efficient routing and comprehensive information retrieval in wireless. In *Parallel and Distributed Processing International Symposium Proceedings , IPDPS 2002*. doi: 10.1109/IPDPS.2002.1016600. 147

Enver Ever. *Performability Modelling of Homogeneous and Heterogeneous Multi-server Systems with Breakdowns and Repairs*. PhD thesis, School of Computing Science Middlesex University, 2007. 4

Enver Ever, R. Luchmun, Leonardo Mostarda, Alfredo Navarra, and P. Shah. Uheed - an unequal clustering algorithm for wireless sensor networks. In *SENSORNETS*, pages 185–193, 2012. ISBN 978-989-8565-01-3. 41, 147

Yue Fang and A.B. McDonald. Dynamic codeword routing (dcr): a cross-layer approach for performance enhancement of general multi-hop wireless routing. In *Sensor and Ad Hoc Communications and Networks, 2004. IEEE SECON 2004. 2004 First Annual IEEE Communications Society Conference on*, pages 255–263. 17

G. Ferrari, P. Medagliani, S. Di Piazza, and M. Martalò. Wireless sensor networks: performance analysis in indoor scenarios. *EURASIP J. Wirel. Commun. Netw.*, (1):41–41, 2007. ISSN 1687-1472. 44

Amer Filipovic and Amitava Datta. Building blocks of energy and cost efficient wireless sensor networks. In Holger Karl, Adam Wolisz, and Andreas Willig, editors, *Wireless Sensor Networks*, volume 2920 of *Lecture Notes in Computer Science*, pages 218–233. Springer Berlin / Heidelberg, 2004. 28, 153

Gerhard Fuchs and Reinhard German. Uml2 activity diagram based programming of wireless sensor networks. In *Proceedings of the 2010 ICSE Workshop on Software Engineering for Sensor Network Applications*, SESENA '10, pages 8–13, New York, NY, USA. ACM. 39

D. Ganesan, B. Krishnamachari, A. Woo, D. Culler, D. Estrin, and S. Wicker. An Empirical Study of Epidemic Algorithms in Large Scale Multihop Wireless Networks, 2008. 5

Tia Gao, C. Pesto, L. Selavo, Yin Chen, JeongGil Ko, Jong Hyun Lim, A. Terzis, A. Watt, J. Jeng, Bor rong Chen, K. Lorincz, and M. Welsh. Wireless medical sensor networks in emergency response: Implementation and pilot results. In *2008 IEEE Conference on Technologies for Homeland Security*, pages 187–192. 23

K. Gill, Shuang-Hua Yang, Fang Yao, and Xin Lu. A zigbee-based home automation system. *IEEE Transactions on Consumer Electronics*, 55(2): 422 –430, may 2009. 94, 120

A. Goldsmith. *Wireless Communications*. Cambridge University Press, 2005. ISBN 9780521837163. URL http://books.google.co.uk/books?id=n-3ZZ9i0s-cC. 29, 87

Bencan Gong, Layuan Li, Shaorong Wang, and Xuejun Zhou. Multihop routing protocol with unequal clustering for wireless sensor networks. In

Proc. of the ISECS Intl Colloquium on Computing, Communication, Control, and Management (CCCM), pages 552–556. IEEE Computer Society, 2008. 131

Reinhard Gotzhein, Marc Krämer, Lothar Litz, and Alain Chamaken. Energy-aware system design with sdl. In *Proceedings of the 14th international SDL conference on Design for motes and mobiles*, SDL'09, pages 19–33, Berlin, Heidelberg, 2009. Springer-Verlag. 36

V.Ç. Güngör and G.P. Hancke. *Industrial Wireless Sensor Networks: Applications, Protocols, and Standards*. Industrial Electronics. Taylor & Francis, 2013. ISBN 9781466500518. URL http://books.google.co.uk/books?id=9YK4x8le-RUC. 5

P. Gupta and P.R. Kumar. The capacity of wireless networks. *Information Theory, IEEE Transactions on*, 46(2):388–404, 2000. ISSN 0018-9448. doi: 10.1109/18.825799. 43

Dae-Man Han and Jae-Hyun Lim. Smart home energy management system using ieee 802.15.4 and zigbee. *IEEE Transactions on Consumer Electronics*, 56(3):1403 –1410, aug. 2010. ISSN 0098-3063. doi: 10.1109/TCE.2010.5606276. 94, 120

W.B. Heinzelman, A.P. Chandrakasan, and H. Balakrishnan. An application-specific protocol architecture for wireless microsensor networks. *IEEE Transactions on Wireless Communications*, 1(4):660–670, 2002. 147

Wendi Rabiner Heinzelman, Anantha Chandrakasan, and Hari Balakrishnan. Energy-efficient communication protocol for wireless microsensor networks. In *Proc. of the 33rd Hawaii Intl Conf. on System Sciences (HICSS)*, Washington, DC, USA, 2000. 131, 139

W.R. Heinzelman, A. Chandrakasan, and H. Balakrishnan. Energy-efficient communication protocol for wireless microsensor networks. In *Proceedings of the 33rd Annual Hawaii International Conference on System Sciences, 2000*, page 10 vol.2. 147

Jason Lester Hill. *System architecture for wireless sensor networks*. PhD thesis, 2003. 36

Robert M. Hirsch, Richard B. Alexander, and Richard A. Smith. Selection of methods for the detection and estimation of trends in water quality. *Water Resources Research*, 27(5):803–813, 1991. 176

Fei Hu and 1972 Cao, Xiaojun. *Wireless sensor networks : principles and practice / Fei Hu, Xiaojun Cao*. Boca Raton, FL : Auerbach Publications, 2010. ISBN 9781420092158 (alk. paper). Formerly CIP. 31

Yih-Chun Hu, D.B. Johnson, and A. Perrig. Sead: secure efficient distance vector routing for mobile wireless ad hoc networks. In *Mobile Computing Systems and Applications, 2002. Proceedings Fourth IEEE Workshop on*, pages 3 – 13, a. doi: 10.1109/MCSA.2002.1017480. 156

Yih-Chun Hu, Adrian Perrig, and David B. Johnson. Ariadne: a secure on-demand routing protocol for ad hoc networks. *Wirel. Netw.*, 11(1-2): 21–38, b. ISSN 1022-0038. 156

Chi-Fu Huang and Yu-Chee Tseng. The coverage problem in a wireless sensor network. In *Proceedings of the 2nd ACM international conference on Wireless sensor networks and applications*, WSNA '03, pages 115–121, New York, NY, USA, 2003. ACM. ISBN 1-58113-764-8. doi: 10.1145/941350.941367. 77

Jean-Pierre Hubaux, Levente Buttyán, and Srdan Capkun. The quest for security in mobile ad hoc networks. In *Proceedings of the 2nd ACM international symposium on Mobile ad hoc networking & computing*, MobiHoc '01, pages 146–155, New York, NY, USA, 2001. ACM. ISBN 1-58113-428-2. 156

M. Imran, A.M. Said, and H. Hasbullah. A survey of simulators, emulators and testbeds for wireless sensor networks. In *International Symposium in Information Technology (ITSim)*, volume 2, pages 897–902, 2010. doi: 10.1109/ITSIM.2010.5561571. 38, 39, 72

Chalermek Intanagonwiwat, Deborah Estrin, Ramesh Govindan, and John Heidemann. Impact of network density on data aggregation in wireless sensor networks. In *Proceedings of the 22 nd International Conference on Distributed Computing Systems (ICDCS'02)*, ICDCS '02, pages 457–, Washington, DC, USA, 2002. IEEE Computer Society. ISBN 0-7695-1585-1. 84

ISO/IEC/IEEE. ISO/IEC/IEEE 42010:2011 Systems and software engineering – Architecture description, 2011. 73, 78

Teerawat Issariyakul and Ekram Hossain. *Introduction to Network Simulator NS2*. Springer Publishing Company, Incorporated, 1 edition, 2008. ISBN 0387717595, 9780387717593. 101

Jacques and Marculescu. Algosensim, December 2011. URL http://tcs.unige.ch/doku.php/code/algosensim/overview. 37, 100

R. Jain. *The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation, and Modeling*. Wiley Professional Computing. Wiley, 1991. ISBN 9780471503361. URL http://books.google.co.uk/books?id=HetQAAAAMAAJ. 3

S. Jardosh and P. Ranjan. A survey: Topology control for wireless sensor networks. In *Signal Processing, Communications and Networking, 2008. ICSCN '08. International Conference on*, pages 422–427. doi: 10.1109/ICSCN.2008.4447231. 6

H. S. Javitz and A. Valdes. The nides statistical component description and justification. *Technical report - Columbia University*, March 1994. 27, 153

Chenjun Jia, Yongjian Liao, and Kangshen Chen. Secure encryption in wireless sensor network. In *Wireless Communications, Networking and Mobile Computing, WiCOM '08. 4th International Conference on*, pages 1–4, 2008. 27, 152

Philo Juang, Hidekazu Oki, Yong Wang, Margaret Martonosi, Li Shiuan Peh, and Daniel Rubenstein. Energy-efficient computing for wildlife tracking:

design tradeoffs and early experiences with zebranet. *SIGOPS Oper. Syst. Rev.*, 36(5):96–107, October 2002. ISSN 0163-5980. 23

H. Karl and A. Willig. *Protocols and Architectures for Wireless Sensor Networks*. Wiley, 2007. ISBN 9780470519233. URL http://books.google.co.uk/books?id=170R-1aZsQYC. 15, 17

Chris Karlof and David Wagner. Secure routing in wireless sensor networks: Attacks and countermeasures. In *In First IEEE International Workshop on Sensor Network Protocols and Applications*, pages 113–127, 2002. 156

S. Khan, A.S.K. Pathan, and N.A. Alrajeh. *Wireless Sensor Networks: Current Status and Future Trends*. Taylor & Francis, 2013. ISBN 9781466588851. URL http://books.google.co.uk/books?id=m8hiNil66HsC. 6

Hyung Seok Kim, Joo-Han Song, and Seok Lee. Energy-efficient traffic scheduling in ieee 802.15.4 for home automation networks. *IEEE Transactions on Consumer Electronics*, 53(2):369 –374, may 2007. 120

Jung-Hwan Kim, Sajjad Hussain Chauhdary, Wen-Cheng Yang, Dong-Sub Kim, and Myong-Soon Park. Produce: A probability-driven unequal clustering mechanism for wireless sensor networks. *Proc. of the 22nd Intl Conf. on Advanced Information Networking and Applications Workshops*, pages 928–933, 2008. 133

Sukun Kim. Wireless sensor networks for structural health monitoring. Technical report, IN UC BERKELEY MASTERâĂŹS THESIS, 2005. 24

JeongGil Ko, Chenyang Lu, M.B. Srivastava, J.A. Stankovic, A. Terzis, and M. Welsh. Wireless sensor networks for healthcare. *Proceedings of the IEEE*, 98(11):1947 –1960, nov. 2010. ISSN 0018-9219. doi: 10.1109/JPROC.2010.2065210. 90

Jiejun Kong, Z. Petros, Haiyun Luo, Songwu Lu, and Lixia Zhang. Providing robust and ubiquitous security support for mobile ad-hoc networks. In

Ninth International Conference on Network Protocols., pages 251 –260, nov. 2001. doi: 10.1109/ICNP.2001.992905. 156

Jiejun Kong, Haiyun Luo, Kaixin Xu, Daniel Lihui Gu, Mario Gerla, and Songwu Lu. Adaptive security for multi-layer ad-hoc networks. In *SPECIAL ISSUE OF WIRELESS COMMUNICATIONS AND MOBILE COMPUTING*, pages 533–547. Wiley Interscience Press, 2002. 156

Chiu-Yuen Koo, Vartika Bhandari, Jonathan Katz, and Nitin H. Vaidya. Reliable broadcast in radio networks: the bounded collision case. In *Proceedings of the twenty-fifth annual ACM symposium on Principles of distributed computing*, PODC '06, pages 258–264, New York, NY, USA, 2006. ACM. ISBN 1-59593-384-0. 155

O. Kosut and Lang Tong. Distributed source coding in the presence of byzantine sensors. *Information Theory, IEEE Transactions on*, 54(6):2550 –2565, june 2008. ISSN 0018-9448. doi: 10.1109/TIT.2008.921867. 155

David Kotz, Calvin Newport, Robert S. Gray, Jason Liu, Yougu Yuan, and Chip Elliott. Experimental evaluation of wireless simulation assumptions. In *MSWiM '04: Proceedings of the 7th ACM international symposium on Modeling, analysis and simulation of wireless and mobile systems*, pages 78–82, New York, NY, USA, 2004. ACM. ISBN 1-58113-953-5. 37, 101, 119

K.Pahlavan and P.Krishnamurthy. *Networking Fundamentals*. John Wiley and Sons, Chichester, UK, 2009. 18, 28, 29, 103, 122

Tronje Krop, Michael Bredel, Matthias Hollick, and Ralf Steinmetz. Jist/mobnet: combined simulation, emulation, and real-world testbed for ad hoc networks. In *Proceedings of the second ACM international workshop on Wireless network testbeds, experimental evaluation and characterization*, WinTECH '07, pages 27–34, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-738-4. 3

P. Kumarawadu, D.J. Dechene, M. Luccini, and A. Sauer. Algorithms for node clustering in wireless sensor networks: A survey. In *4th International*

Conference on Information and Automation for Sustainability. ICIAFS 2008., pages 295–300, 2008. doi: 10.1109/ICIAFS.2008.4783999. 6

Mauri Kuorilehto, Marko Hännikäinen, and Timo D. Hämäläinen. A survey of application distribution in wireless sensor networks. *EURASIP J. Wirel. Commun. Netw.*, 2005(5):774–788, October 2005. ISSN 1687-1472. 54

K. Lahmar, R. Cheour, and M. ABID. Wireless sensor networks: Trends, power consumption and simulators. In *Modelling Symposium (AMS), 2012 Sixth Asia*, pages 200–204, May 2012. doi: 10.1109/AMS.2012.50. 99

L. Lamport. The byzantine generals problem. *ACM Transactions on Programming Languages and Systems*, 4:382–401, 1982. 154, 155, 159

K. Langendoen, W. Hu, F. Ferrari, M. Zimmerling, and L. Mottola. *Real-World Wireless Sensor Networks: Proceedings of the 5th International Workshop, REALWSN 2013, Como (Italy), September 19-20, 2013*. Lecture Notes in Electrical Engineering. Springer International Publishing, 2014. ISBN 9783319030708. URL http://books.google.co.uk/books?id=Tb4CngEACAAJ. 2, 170

BenoÃőt LatrÃľ, Pieter De Mil, Ingrid Moerman, Niek Van Dierdonck, Bart Dhoedt, and Piet Demeester. Maximum throughput and minimum delay in ieee 802.15.4. In *MSN*, volume 3794 of *Lecture Notes in Computer Science*, pages 866–876. Springer, 2005. ISBN 3-540-30856-3. 54, 145

Averill M. Law and David W. Kelton. *Simulation Modelling and Analysis*. McGraw-Hill Education - Europe, April 2000. ISBN 0071165371. URL http://www.worldcat.org/isbn/0071165371. 4

Sang Hyuk Lee, Soobin Lee, Heecheol Song, and Hwang Soo Lee. Wireless sensor network design for tactical military applications: remote large-scale environments. In *Proceedings of the 28th IEEE conference on Military communications*, MILCOM'09, pages 911–917, Piscataway, NJ, USA, 2009. IEEE Press. ISBN 978-1-4244-5238-5. 22

Philip Levis, Nelson Lee, Matt Welsh, and David Culler. Tossim: accurate and scalable simulation of entire tinyos applications. In *Proceedings of the 1st international conference on Embedded networked sensor systems*, SenSys '03, pages 126–137, New York, NY, USA, 2003. ACM. ISBN 1-58113-707-9. 37, 99

Chengfa Li, Mao Ye, Guihai Chen, and Jie Wu. An energy-efficient unequal clustering mechanism for wireless sensor networks. In *Mobile Adhoc and Sensor Systems Conference, 2005. IEEE International Conference on*, pages 8 pp.–604, 2005a. doi: 10.1109/MAHSS.2005.1542849. 26, 131

Chengfa Li, Mao Ye, Guihai Chen, and Jie Wu. An energy-efficient unequal clustering mechanism for wireless sensor networks. In *Proc. of the IEEE Intl Conf. on Mobile Adhoc and Sensor Systems Conf. (MASS)*, 2005b. 133

Mo Li. A survey on topology issues in wireless sensor network, 2006. 6

Zhenjiang Li and Yunhao Liu. Underground structure monitoring with wireless sensor networks. In *6th International Symposium on Information Processing in Sensor Networks*, pages 69–78, 2007. doi: 10.1109/IPSN.2007.4379666. 24

Chih-Kuang Lin, Vladimir Zadorozhny, and Prashant Krishnamurthy. Efficient hybrid channel access for data intensive sensor networks. In *Proceedings of the 21st International Conference on Advanced Information Networking and Applications Workshops - Volume 02*, AINAW '07, pages 659–664, Washington, DC, USA, 2007. IEEE Computer Society. ISBN 0-7695-2847-3. 30, 31

S. Lindsey and C.S. Raghavendra. Pegasis: Power-efficient gathering in sensor information systems. In *Aerospace Conference Proceedings, 2002. IEEE*, pages 3–1125–3–1130. doi: 10.1109/AERO.2002.1035242. 147

D. Liu and P. Ning. *Security for wireless sensor networks.* Advances in Information Security, 28. Springer Science+Business Media, LLC, 2007.

ISBN 9780387467818. URL http://books.google.co.uk/books?id=eCqPK9p9Z1IC. 26

J. López and J. Zhou. *Wireless Sensor Network Security*. Cryptology and information security series. IOS Press, 2008. ISBN 9781586038137. URL http://books.google.co.uk/books?id=pA2XUtdwewAC. 152

K. Lorincz, D.J. Malan, T.R.F. Fulford-Jones, A. Nawoj, A. Clavel, V. Shnayder, G. Mainland, M. Welsh, and S. Moulton. Sensor networks for emergency response: challenges and opportunities. *Pervasive Computing, IEEE*, 3(4):16 – 23, oct.-dec. 2004. ISSN 1536-1268. doi: 10.1109/MPRV.2004.18. 22, 79, 88

Fernando Losilla, Cristina Vicente-Chicote, Bárbara Álvarez, Andrés Iborra, and Pedro Sánchez. Wireless sensor network application development: an architecture-centric mde approach. In *Proceedings of the First European conference on Software Architecture*, ECSA'07, pages 179–194, Berlin, Heidelberg, 2007. Springer-Verlag. ISBN 3-540-75131-9, 978-3-540-75131-1. 36

Haiyun Luo, P. Zerfos, Jiejun Kong, Songwu Lu, and Lixia Zhang. Self-securing ad hoc wireless networks. In *Seventh International Symposium on Computers and Communications.*, pages 567 – 574, july 2002. doi: 10.1109/ISCC.2002.1021731. 156

Jianqing Ma, Ping Yi, Yiping Zhong, and Shiyong Zhang. Sfirewall: A firewall in wireless sensor networks. In *WiCOM 2006.International Conference on Wireless Communications, Networking and Mobile Computing*, pages 1 –4, sept. 2006. doi: 10.1109/WiCOM.2006.280. 27, 152

Alan Mainwaring, David Culler, Joseph Polastre, Robert Szewczyk, and John Anderson. Wireless sensor networks for habitat monitoring. In *Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications*, WSNA '02, pages 88–97, New York, NY, USA, 2002. ACM. ISBN 1-58113-589-0. 23, 50

David Malan, Thaddeus Fulford-jones, Matt Welsh, and Steve Moulton. Codeblue: An ad hoc sensor network infrastructure for emergency medical care. In *In International Workshop on Wearable and Implantable Body Sensor Networks*, 2004. 23, 88

I. Malavolta, H. Muccini, P. Pelliccione, and D.A. Tamburri. Providing architectural languages and tools interoperability through model transformation technologies. *IEEE Transactions on Software Engineering*, 36(1): 119 –140, jan.-feb. 2010. ISSN 0098-5589. 82

D. Malki and M. Reiter. A high-throughput secure reliable multicast protocol. In *Computer Security Foundations Workshop, 9th IEEE Proceedings*, pages 9 –17, jun 1996. doi: 10.1109/CSFW.1996.503686. 154

Arati Manjeshwar and D.P. Agrawal. Teen: a routing protocol for enhanced efficiency in wireless sensor networks. In *15th Internationa Parallel and Distributed Processing Symposium Proceedings l*, pages 2009–2015, 2001. doi: 10.1109/IPDPS.2001.925197. 147

Marco Martalò, Stefano Busanelli, and Gianluigi Ferrari. Markov chain-based performance analysis of multihop ieee 802.15.4 wireless networks. *Perform. Eval.*, 66(12):722–741. 44

S.P. Millard, N.K. Neerchal, and P. Dixon. *Environmental Statistics with S-PLUS*. Chapman & Hall/CRC Applied Environmental Statistics. Taylor & Francis, 2000. ISBN 9780849371684. URL http://books.google.co.in/books?id=7peG3cvtDpwC. 176

A. F. Mini, Badri Nath, and Antonio A. F. Loureiro. A probabilistic approach to predict the energy consumption in wireless sensor networks. In *In IV Workshop de Comunicao sem Fio e Computao Mvel. So Paulo*, pages 23–25, 2002. 43

A. Mishra, K. Nadkarni, and A. Patcha. Intrusion detection in wireless ad hoc networks. *Wireless Communications, IEEE*, 11(1):48 – 60, feb 2004. ISSN 1536-1284. doi: 10.1109/MWC.2004.1269717. 154

J. Misic and R. Udayshankar. Slave-slave bridging in 802.15.4 beacon enabled networks. In *Wireless Communications and Networking Conference, 2007.WCNC 2007. IEEE*, pages 3890–3895. doi: 10.1109/WCNC.2007.711. 44

J. Misic, J. Fung, and V.B. Misic. Interconnecting 802.15.4 clusters in master-slave mode: queueing theoretic analysis. In *ISPAN 2005. Proceedings. 8th International Symposium on Parallel Architectures,Algorithms and Networks,.*, pages 8 pp.–, 2005. doi: 10.1109/ISPAN.2005.53. 44

J. Misic, S. Shafi, and V.B. Misic. Performance of a beacon enabled ieee 802.15.4 cluster with downlink and uplink traffic. *Parallel and Distributed Systems, IEEE Transactions on*, 17(4):361–376, 2006. ISSN 1045-9219. 44

Leonardo Mostarda and Alfredo Navarra. Distributed intrusion detection systems for enhancing security in mobile wireless sensor networks. *IJDSN*, 4(2):83–109, 2008. 156

Luca Mottola and Gian Pietro Picco. Programming wireless sensor networks: Fundamental concepts and state of the art. *ACM Comput. Surv.*, 43(3): 19:1–19:51, April 2011. ISSN 0360-0300. 36, 38, 72

M. M R Mozumdar, F. Gregoretti, L. Lavagno, L. Vanzago, and S. Olivieri. A framework for modeling, simulation and automatic code generation of sensor network application. In *Sensor, Mesh and Ad Hoc Communications and Networks, 2008. SECON '08. 5th Annual IEEE Communications Society Conference on*, pages 515–522, 2008. doi: 10.1109/SAHCN.2008.68. 36, 39

Calvin Newport, David Kotz, Yougu Yuan, Robert S. Gray, Jason Liu, and Chip Elliott. Experimental evaluation of wireless simulation assumptions. pages 78–82. ACM Press, 2004. 53

FredrickA. Omondi, Enver Ever, Purav Shah, and Orhan Gemikonakli. Modelling wireless sensor networks for performability evaluation. In Jacek CichoÅĎ, Maciej GÈľbala, and Marek Klonowski, editors, *Ad-hoc, Mobile,*

and Wireless Network, volume 7960 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2013a. 144, 146

FredrickA. Omondi, Enver Ever, Purav Shah, and Orhan Gemikonakli. Modelling wireless sensor networks for performability evaluation. In Jacek CichoÅĎ, Maciej GÈľbala, and Marek Klonowski, editors, *Ad-hoc, Mobile, and Wireless Network*, volume 7960 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2013b. 41

K. Pahlavan and P. Krishnamurthy. *Networking Fundamentals: Wide, Local and Personal Area Communications*. Wiley, 2009. ISBN 9780470779439. URL http://books.google.co.uk/books?id=WOCrSSfxE-EC. 86, 87, 92, 103

M. Pallikonda Rajasekaran, S. Radhakrishnan, and P. Subbaraj. Sensor grid applications in patient monitoring. *Future Gener. Comput. Syst.*, 26(4): 569–575, April 2010. ISSN 0167-739X. 88

Panagiotis Papadimitratos and Zygmunt Haas. Secure routing for mobile ad hoc networks. *MOBILE COMPUTING AND COMMUNICATIONS REVIEW*, 1(2):27–31, 2002. 154, 156

Andrzej Pelc and David Peleg. Feasibility and complexity of broadcasting with random transmission failures. *Theor. Comput. Sci.*, 370(1-3):279–292, February 2007. ISSN 0304-3975. 155

Gian Pietro Picco. Software engineering and wireless sensor networks: happy marriage or consensual divorce? In *Proceedings of the FSE/SDP workshop on Future of software engineering research*, FoSER '10, pages 283–286, New York, NY, USA, 2010. ACM. ISBN 978-1-4503-0427-6. 38, 73, 79

J. Polley, D. Blazakis, J. McGee, D. Rusk, and J.S. Baras. Atemu: a fine-grained sensor network simulator. In *First Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks. IEEE SECON*, pages 145 – 152, oct. 2004. doi: 10.1109/SAHCN.2004.1381912. 37, 99

Daniele Puccinelli and Martin Haenggi. Wireless sensor networks: applications and challenges of ubiquitous sensing. *IEEE Circuits and Systems Magazine*, 5, 2005. 21

Tie Qiu, Lin Feng, Feng Xia, Guowei Wu, and Yu Zhou. A packet buffer evaluation method exploiting queueing theory for wireless sensor networks. *Comput. Sci. Inf. Syst.*, 8(4):1028–1049, 2011. 43

R. Rajagopalan and P.K. Varshney. Data-aggregation techniques in sensor networks: a survey. *Communications Surveys Tutorials, IEEE*, 8(4):48–63, 2006. ISSN 1553-877X. doi: 10.1109/COMST.2006.283821. 84

Venkatesh Rajendran, Katia Obraczka, and J. J. Garcia-Luna-Aceves. Energy-efficient collision-free medium access control for wireless sensor networks. In *Proceedings of the 1st international conference on Embedded networked sensor systems*, SenSys '03, pages 181–192, New York, NY, USA, 2003. ACM. ISBN 1-58113-707-9. 31

Iyappan Ramachandran, Arindam K. Das, and Sumit Roy. Analysis of the contention access period of IEEE 802.15.4 MAC. *ACM Trans. Sen. Netw.*, 3(1):4, 2007. ISSN 1550-4859. 44

T.S. Rappaport. *Wireless communications: principles and practice*. Prentice Hall communications engineering and emerging technologies series. Prentice Hall PTR, 1996. ISBN 9780133755367. URL `http://books.google.co.uk/books?id=C_pSAAAAMAAJ`. 29, 30, 86, 87

Peng Ren, Jiansheng Qian, Leida Li, Zhikai Zhao, and Xiaobin Li. Unequal clustering scheme based leach for wireless sensor networks. In *Genetic and Evolutionary Computing (ICGEC), 2010 Fourth International Conference on*, pages 90–93, Dec 2010a. doi: 10.1109/ICGEC.2010.30. 133

Peng Ren, Jiansheng Qian, Leida Li, Zhikai Zhao, and Xiaobin Li. Unequal clustering scheme based leach for wireless sensor networks. In *Proc. of the 4th Intl Conf. on Genetic and Evolutionary Computing (ICGEC)*, 2010b. 139, 140

Rodrigo Roman. Applying intrusion detection systems to wireless sensor networks. In *in CCNC 2006: Proceeding of the 3rd IEEE Consumer Communications and Networking Conference*, pages 640–644, 2006. 152

K. Romer and F. Mattern. The design space of wireless sensor networks. *Wireless Communications, IEEE*, 11(6):54–61, 2004a. ISSN 1536-1284. doi: 10.1109/MWC.2004.1368897. 38

K. Romer and F. Mattern. The design space of wireless sensor networks. *Wireless Communications, IEEE*, 11(6):54 – 61, dec. 2004b. ISSN 1536-1284. 72

Giovanni Russello, Leonardo Mostarda, and Naranker Dulay. A policy-based publish/subscribe middleware for sense-and-react applications. *Journal of Systems and Software*, 84(4):638–654, 2011. 161

Paolo Santi. Topology control in wireless ad hoc and sensor networks. *ACM Comput. Surv.*, 37(2):164–194, June 2005. ISSN 0360-0300. 6

K. Sanzgiri, B. Dahill, B.N. Levine, C. Shields, and E.M. Belding-Royer. A secure routing protocol for ad hoc networks. In *10th IEEE International Conference on Network Protocols, Proceedings*, pages 78 – 87, Nov. 2002. doi: 10.1109/ICNP.2002.1181388. 156

Curt Schurgers, Vlasios Tsiatsis, Saurabh Ganeriwal, and Mani Srivastava. Topology management for sensor networks: Exploiting latency and density. pages 135–145, 2002. 42

John S Seybold. *Introduction to RF Propagation*. Wiley, Newark, NJ, 2005. 87

S. Shakkottai, R. Srikant, and N. Shroff. Unreliable sensor grids: coverage, connectivity and diameter. In *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies*, volume 2, pages 1073–1083 vol.2, 2003. 43

E. Shi and A. Perrig. Designing secure sensor networks. *Wireless Communications, IEEE*, 11(6):38 – 43, dec. 2004. ISSN 1536-1284. doi: 10.1109/MWC.2004.1368895. 155

Ryo Shimizu, Kenji Tei, Yoshiaki Fukazawa, and Shinichi Honiden. Model driven development for rapid prototyping and optimization of wireless sensor network applications. In *Proceedings of the 2nd Workshop on Software Engineering for Sensor Network Applications*, SESENA '11, pages 31–36, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0583-9. 37

Victor Shnayder, Bor-rong Chen, Konrad Lorincz, Thaddeus R. F. Fulford Jones, and Matt Welsh. Sensor networks for medical care. In *Proceedings of the 3rd international conference on Embedded networked sensor systems*, SenSys '05, pages 314–314, New York, NY, USA, 2005. ACM. ISBN 1-59593-054-X. 22, 88

A. Shrestha and Liudong Xing. A performance comparison of different topologies for wireless sensor networks. In *IEEE Conference on Technologies for Homeland Security,*, pages 280–285, 2007. doi: 10.1109/THS.2007.370059. 24

Suresh Singh and C. S. Raghavendra. Pamas—power aware multi-access protocol with signalling for ad hoc networks. *SIGCOMM Comput. Commun. Rev.*, 28(3):5–26, July 1998. ISSN 0146-4833. doi: 10.1145/293927.293928. 42

K. Sohraby, D. Minoli, and T. Znati. *Wireless Sensor Networks: Technology, Protocols, and Applications*. Wiley, 2007. ISBN 9780470112755. URL http://books.google.co.uk/books?id=aEPBTmUhyI0C. 24

John A. Stankovic. Research challenges for wireless sensor networks. *SIGBED Rev.*, 1(2):9–12, July 2004. ISSN 1551-3688. 36

T. Stathopoulos, R. Kapur, D. Estrin, J. Heidemann, and Lixia Zhang. Application-based collision avoidance in wireless sensor networks. In *29th Annual IEEE International Conference on Local Computer Networks*, pages 506 – 514, nov. 2004. 31

M. Stillerman, C. Marceau, and M. Stillman. Intrusion detection for distributed applications. *Communications of the ACM*, July 1999. 27, 153

Bo Sun, L. Osborne, Yang Xiao, and S. Guizani. Intrusion detection techniques in mobile ad hoc and wireless sensor networks. *Wireless Communications, IEEE*, 14(5):56 –63, october 2007. ISSN 1536-1284. doi: 10.1109/MWC.2007.4396943. 154

Y. Tachwali, H. Refai, and J.E. Fagan. Minimizing hvac energy consumption using a wireless sensor network. In *Industrial Electronics Society. IECON 2007. 33rd Annual Conference of the IEEE*, pages 439 –444, nov. 2007. doi: 10.1109/IECON.2007.4460329. 120

M. Tavakoli, L. Turicchia, and R. Sarpeshkar. An ultra-low-power pulse oximeter implemented with an energy-efficient transimpedance amplifier. *Biomedical Circuits and Systems, IEEE Transactions on*, 4(1):27 –38, feb. 2010. ISSN 1932-4545. doi: 10.1109/TBCAS.2009.2033035. 91

S. T.Eckmann, G. Vigna, and R. A. Kemmer. Statl: An attack language for state-based intrusion detection. *Journal of Computer Security*, 10:71–104, 2002. 27, 152

Sameer Tilak, Nael B. Abu-Ghazaleh, and Wendi Heinzelman. A taxonomy of wireless micro-sensor network models. *ACM MOBILE COMPUTING AND COMMUNICATIONS REVIEW*, 6:28–36, 2002. 20, 45

Ben L. Titzer and et al. Avrora: Scalable sensor network simulation with precise timing. In *IN PROC. OF THE 4TH INTL. CONF. ON INFORMATION PROCESSING IN SENSOR NETWORKS (IPSN*, pages 477–482, 2005. 37, 99

Kishor S. Trivedi. *Probability and statistics with reliability, queuing and computer science applications*. John Wiley and Sons Ltd., Chichester, UK, 2nd edition edition, 2002. ISBN 0-471-33341-7. 3

H. Vaccaro and G. Liepins. Detection of anomalous computer session activity. *In proc. of the 1989 Synopsium on Security and privacy*, (1-3):280–289, May 1989. 27, 153

Tijs van Dam and Koen Langendoen. An adaptive energy-efficient mac protocol for wireless sensor networks. In *Proceedings of the 1st international conference on Embedded networked sensor systems*, SenSys '03, pages 171–180, New York, NY, USA, 2003. ACM. ISBN 1-58113-707-9. 51, 52, 85

L. Van Hoesel, T. Nieberg, Jian Wu, and P. J M Havinga. Prolonging the lifetime of wireless sensor networks by cross-layer interaction. *Wireless Communications, IEEE*, 11(6):78–86, 2004. ISSN 1536-1284. doi: 10.1109/MWC.2004.1368900. 17

Cristina Vicente-Chicote, Fernando Losilla, BÃąrbara ÃĄlvarez, AndrÃľs Iborra, and Pedro SÃąnchez. Applying mde to the development of flexible and reusable wireless sensor networks. *Int. J. Cooperative Inf. Syst.*, 16 (3/4):393–412, 2007. 39

Yong Wang, G. Attebury, and B. Ramamurthy. A survey of security issues in wireless sensor networks. *Communications Surveys Tutorials, IEEE*, 8 (2):2–23, 2006. ISSN 1553-877X. doi: 10.1109/COMST.2006.315852. 17

Yunbo Wang, Mehmet C Vuran, and Steve Goddard. Cross-layer analysis of the end-to-end delay distribution in wireless sensor networks. *IEEE/ACM Transactions on Networking (TON)*, 20(1):305–318, 2012. 40, 41, 50

G. Werner-Allen, K. Lorincz, M. Ruiz, O. Marcillo, J. Johnson, J. Lees, and M. Welsh. Deploying a wireless sensor network on an active volcano. *Internet Computing, IEEE*, 10(2):18–25, 2006. ISSN 1089-7801. doi: 10.1109/MIC.2006.26. 23

M. B. Wilk and R. Gnanadesikan. Probability plotting methods for the analysis of data. *Biometrika*, 55(1):1–17, March 1968. ISSN 0006-3444. 176

A. Willig. Wireless sensor networks: concept, challenges and approaches. *Elektrotechnik & Informationstechnik*, 123(6):224 –231, June 2006a. 73

Andreas Willig. Wireless sensor networks: concept, challenges and approaches. *Elektrotechnik und Informationstechnik*, pages 21–31, 2006b. 38

Ning Xu, Sumit Rangwala, Krishna Kant Chintalapudi, Deepak Ganesan, Alan Broad, Ramesh Govindan, and Deborah Estrin. A wireless sensor network for structural monitoring. In *Proceedings of the 2nd international conference on Embedded networked sensor systems*, SenSys '04, pages 13–24, New York, NY, USA, 2004. ACM. ISBN 1-58113-879-2. 24

Chen Xuhui, Yang Zhiming, and Cheng Huiyan. Unequal clustering mechanism of leach protocol for wireless sensor networks. In *Proceedings of the WRI World Congress on Computer Science and Information Engineering - Volume 01*, CSIE '09, pages 258–262, Washington, DC, USA, 2009. IEEE Computer Society. ISBN 978-0-7695-3507-4. 26, 133

Wei Ye, John Heidemann, and Deborah Estrin. An energy-efficient mac protocol for wireless sensor networks. In *Proceedings of the IEEE Infocom*, pages 1567–1576, New York, NY, USA, June 2002. IEEE. 42

Wei Ye, J. Heidemann, and D. Estrin. Medium access control with coordinated adaptive sleeping for wireless sensor networks. *Networking, IEEE/ACM Transactions on*, 12(3):493 – 506, june 2004a. 102

Wei Ye, John Heidemann, and Deborah Estrin. Medium access control with coordinated adaptive sleeping for wireless sensor networks. *IEEE/ACM Trans. Netw.*, 12(3):493–506, June 2004b. ISSN 1063-6692. 85, 90

Ossama Younis and Sonia Fahmy. Distributed clustering in ad-hoc sensor networks: A hybrid, energy-efficient approach. In *Proc. of the 23rd IEEE Intl Conf. on Computer Communications INFOCOM*, 2004a. 147

Ossama Younis and Sonia Fahmy. Heed: A hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks. *IEEE Transac-*

tions on Mobile Computing, 3:366–379, 2004b. 128, 129, 133, 134, 135, 136, 138, 149

Chiu yuen Koo. Broadcast in radio networks tolerating byzantine adversarial behavior. In *In PODC âĂŹ04: Proceedings of the twenty-third annual ACM symposium on Principles of distributed computing*, pages 275–282. ACM Press, 2004. 155

Amir Sepasi Zahmati, Bahman Abolhassani, Ali Asghar, Beheshti Shirazi, and Ali Shojaee Bakhtiari. An energy-efficient protocol with static clustering for wireless sensor networks. 147

Manel Guerrero Zapata. Secure ad hoc on-demand distance vector routing. *SIGMOBILE Mob. Comput. Commun. Rev.*, 6(3):106–107, June 2002. ISSN 1559-1662. 156

Yuhong Zhang and Wei Li. An energy-based stochastic model for wireless sensor networks. *Wireless Sensor Network*, 3(9):322–328, 2011. 41

Yuhong Zhang and Wei Wayne Li. Modeling and energy consumption evaluation of a stochastic wireless sensor network. *EURASIP J. Wireless Comm. and Networking*, page 282, 2012. 41

Xinyuan Zhao and Neng Wang. An unequal layered clustering approach for large scale wireless sensor networks. In *2nd International Conference on Future Computer and Communication (ICFCC)*, volume 1, pages V1–750–V1–756, 2010a. doi: 10.1109/ICFCC.2010.5497328. 26, 133

Xinyuan Zhao and Neng Wang. An unequal layered clustering approach for large scale wireless sensor networks. In *Proc. of the 2nd Intl Conf. on Future Computer and Communication (ICFCC)*, 2010b. 133

J. Zheng and A. Jamalipour. *Wireless Sensor Networks: A Networking Perspective*. Wiley, 2009. ISBN 9780470443514. URL `http://books.google.co.uk/books?id=qOPk-NWkgiMC`. 33

Rong Zheng, Jennifer C. Hou, and Lui Sha. Asynchronous wakeup for ad hoc networks. In *Proceedings of the 4th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, MobiHoc '03, pages 35–45, New York, NY, USA, 2003. ACM. ISBN 1-58113-684-6. 42

Gang Zhou, Tian He, Sudha Krishnamurthy, and John A. Stankovic. Models and solutions for radio irregularity in wireless sensor networks. *ACM Trans. Sen. Netw.*, 2(2):221–262, May 2006. ISSN 1550-4859. 30

Lidong Zhou and Zygmunt J. Haas. Securing ad hoc networks. *IEEE NETWORK MAGAZINE*, 13:24–30, 1999. 156

M. Zuniga and B. Krishnamachari. Analyzing the transitional region in low power wireless links. In *First Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks, IEEE SECON.*, pages 517 – 526, Oct. 2009. doi: 10.1109/SAHCN.2004.1381954. 5, 53, 119